

Aplicación de técnicas de clusterización como inicializadores y la metaheurística ILS, al Problema de Ruteo Vehículos con Múltiples Depósitos (MDVRP)

Jorge Andrés Garcés Bermúdez

Proyecto de grado presentado como requisito
parcial para optar al título de
Magister en Ingeniería Eléctrica

Pereira,
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
Maestría en Ingeniería Eléctrica.



Aplicación de técnicas de clusterización como inicializadores y la metaheurística ILS, al Problema de Ruteo Vehículos con Múltiples Depósitos (MDVRP)

©Jorge Andrés Garcés Bermúdez

Director: Ramón Alfonso Gallego Rendón

Pereira,

Programa de Maestría en Ingeniería Eléctrica.

Universidad Tecnológica de Pereira

La Julita. Pereira(Colombia)

TEL: (+57)(6)3137122

www.utp.edu.co

Versión web disponible en: *<http://recursosbiblioteca.utp.edu.co/tesisd/index.html>*

Dedicatoria

A mi madre. El ser máspreciado en mi vida.

A mi novia por su apoyo y sabios consejos.

A los ingenieros Mario Aldemar Ríos Giraldo y Mario Fernando Ramírez Lozano, de la Empresa de Energía del Quindío, por su colaboración y tiempo dado para este proyecto.

Al ingeniero Ramón Alfonso Gallego Rendón por su gran apoyo, motivación y orientación para el desarrollo de esta tesis.

Resumen

En este Proyecto de Grado se propone una metodología híbrida que combina las técnicas aglomerativas de Clústerización para generar soluciones iniciales con un algoritmo de búsqueda local iterada, *iterated location search* (ILS) para resolver el problema de ruteo de vehículos considerando múltiples depósitos (MDVRP).

El MDVRP es clasificado como NP duro, en el que se atiende a un conjunto de clientes con una demanda determinada. La función objetivo del problema consiste en minimizar el total de la distancia recorrida por las rutas, teniendo en cuenta que todos los clientes deben ser atendidos cumpliendo restricciones de capacidad de depósitos y vehículos.

En la etapa de inicialización se proponen algoritmos heurísticos basados en métodos de clusterización y cuyo objetivo es identificar una estructura inicial, para ser usadas como soluciones de inicio en el proceso de optimización. Las respuestas identificadas son de excelente calidad y para obtenerlas se requiere de bajos tiempos de cómputo. En la etapa de optimización se utiliza el algoritmo ILS el cual es implementado con base en conceptos de intensificación y diversificación y para esto se usa operadores de tipo inter e intra ruta. Además, y con el fin de permitir que el procedimiento viaje entre las regiones infactibles, son aceptadas sobrecargas en vehículos, depósitos y excesos en la longitud de las rutas. Estas violaciones de los límites, denominadas infactibilidades, son llevadas en cuenta en la función objetivo como penalidades.

Para validar la metodología presentada en esta tesis se utilizan instancias diseñadas en la

literatura [15]. Los resultados obtenidos muestran un buen desempeño.

Tabla de Contenido

1. Introducción	1
1.1. Planteamiento del problema	2
1.2. Objetivos	3
1.2.1. General	3
1.2.2. Específicos	3
1.3. Estado del arte	3
1.4. Principales resultados	8
1.5. Estructura del trabajo de grado	8
2. Ruteo de Vehículos con Múltiples Depósitos (MDVRP)	9
2.1. Marco conceptual	10
2.2. Modelo Matemático	12
2.2.1. Nomenclatura	13
2.2.2. Estructura matemática del modelo	14
3. Método de solución	16
3.1. Técnicas de Clústerización	16
3.1.1. Distancia Mínima (SL)	19
3.1.2. Distancia Máxima (CL)	20
3.1.3. Distancia Promedio no Ponderada (UAL)	20
3.1.4. Distancia Promedio Ponderada (AL)	20
3.1.5. Método del Centroide Ponderado (WCM)	20

3.1.6. Método de la Mediana (CMU)	21
3.1.7. Método de Ward (W)	21
3.2. Búsqueda Local Iterada (ILS)	23
3.3. Operadores para la construcción de esquemas de vecindad	26
3.3.1. Shift(1,0):	26
3.3.2. Swap(1,1):	27
3.3.3. Shift(2,0):	27
3.3.4. Swap(2,1):	28
3.3.5. Swap(2,2):	28
3.3.6. 2-OPT:	29
3.3.7. Split:	29
3.4. Metodología propuesta	30
3.5. Procedimiento de Clústerización	31
3.5.1. Procedimiento para generar la solución inicial	32
3.6. Algoritmo de Búsqueda Local Iterada (ILS)	34
3.6.1. Procedimiento de perturbación en el algoritmo ILS	36
4. Análisis de resultados	37
4.1. Resultados etapa de inicialización	38
4.2. Resultados del algoritmo ILS usando instancias tipo P para el MDVRP . . .	42
4.3. Resultados del algoritmo ILS para instancias del MDVRP con restricción de distancia	43
5. Conclusiones y trabajo futuro	45

Capítulo 1

Introducción

Las empresas de distribución de bienes buscan ofrecer un buen servicio, que consiste en una adecuada asistencia y bajos costos, dentro de lo anterior se destaca la necesidad de realizar la programación de las rutas de los vehículos a mínimo costo y que cumpla las restricciones operativas.

El Problema de Ruteo de Vehículos considerando Múltiples Depósitos (MDVRP), es un caso en el que existen varios depósitos (cada uno con una flota de vehículos independiente) que debe servir a todos los clientes, se describe matemáticamente mediante una formulación lineal entera mixta (MILP), y puede ser definido de la siguiente manera: dado un conjunto de depósitos I y de clientes J , donde el objetivo es definir una secuencia de entrega de bienes y servicios, se minimiza la distancia o el tiempo recorrido necesario para atender la demanda D_j de todos los clientes, por medio de vehículos idénticos con capacidad Q_k . Se deben cumplir los siguientes requisitos en el objetivo alcanzado: i) A un cliente se le da servicio exactamente una vez; ii) Cada vehículo arranca y termina en el depósito; iii) La demanda total de cualquier ruta no debe exceder la capacidad de vehículo.

El MDVRP se resuelve a través de técnicas exactas, heurísticas o metaheurísticas y en algunos casos la combinación de ambas metodologías denominadas Matheurísticas. Este problema genera una explosión combinatorial que se refleja en los altos tiempos computacionales para obtener la solución. Adicionalmente, dichos tiempos crecen exponencialmente con el tamaño del problema, razón por la cual, en términos de la

complejidad computacional se clasifica dentro de la categoría NP-duro. Lo anterior, explica el interés creciente en la utilización de las técnicas heurísticas y metaheurísticas, que si bien no tienen un sustento matemático como las técnicas exactas, su uso se justifica por la calidad y tiempo de los resultados obtenidos.

Las principales contribuciones de esta tesis son: 1) Estudio de diferentes estrategias de clusterización para generar soluciones iniciales para un algoritmo ILS (Iterated Local Search) aplicado en la solución del MDVRP. 2) Implementación del algoritmo ILS haciendo uso de estrategias de intensificación y diversificación, las cuales usan operadores inter e intra-ruta, como también conceptos de clusterización.

1.1. Planteamiento del problema

El ruteo de vehículos (VRP) es un problema clásico de la investigación de operaciones debido a su alta influencia económica y dificultad para ser resuelto. La solución de este problema juega un papel importante en la cadena de suministro de las empresas cuyas actividades involucran transporte de bienes y servicios. Generalmente las empresas se ven obligadas a enfrentar este tipo de problemas dentro de su propio sistema de distribución, buscando con esto adecuados modos de transporte, lo que genera ahorros económicos en la compañía. En consecuencia a lo anterior, han surgido diferentes variantes del VRP, como es el caso del MDVRP, que es de gran aplicación en empresas de distribución de bienes y servicios, así como en las de logística que tienen más de un depósito para atender a sus clientes, en el cual se cuenta con m depósitos y k vehículos para atenderlos. Cada cliente debe ser atendido por un único vehículo, el cual debe empezar y finalizar su ruta en el mismo depósito. Como dato de entrada se tienen la capacidad de vehículos, depósitos y demanda de los clientes, además de las coordenadas en el espacio (x,y) de los puntos de depósitos y clientes. Los despachos desde los depósitos no deben exceder la capacidad disponible y las rutas programadas no deben exceder la capacidad del vehículo asignado.

El problema de distribuir productos desde ciertos depósitos a sus usuarios finales juega un papel central en la gestión de algunos sistemas logísticos, y su adecuada planificación puede significar considerables ahorros. Esos potenciales ahorros justifican en gran medida la

utilización de técnicas de investigación operativa como facilitadoras de la planificación, dado que se estima que los costos de transporte representan entre el 10 % y el 30 % del costo final de los bienes.

1.2. Objetivos

1.2.1. General

Desarrollar un algoritmo metaheurístico ILS (Iterated Local Search) para solucionar el problema de ruteo de vehículos multi-depósito utilizando técnicas de Clústerización .

1.2.2. Específicos

- Recopilar información en base de datos disponibles acerca del ruteo de vehículos multi-depósito, modelos propuestos por otros autores necesarios para alcanzar el estado del arte.
- Estudiar el modelo matemático del MDVRP.
- Estudiar técnicas de Clústerización y algoritmo ILS.
- Plantear una metodología para solucionar el MDVRP utilizando técnicas de Clústerización y la metaheurística ILS.
- Implementar la metodología en C++.
- Verificar la metodología con diferentes instancias de la literatura.
- Documentar los resultados de la investigación.

1.3. Estado del arte

El problema de ruteo de vehículos fue propuesto por en el año 1959 por Dantzing y Ramser [1] para el problema de ruteo de carro-tanques de combustible en las estaciones

de servicio. A partir de dicha formulación surgen otras versiones que involucran múltiples depósitos MDVRP. En el año 2015, Montoya et.al [2] realiza una revisión bibliográfica para el MDVRP, en donde encuentra un crecimiento de los artículos publicados hasta el año 2014, aquí se extiende la revisión hasta el año 2017 y es ilustrado en la Figura 1.1.

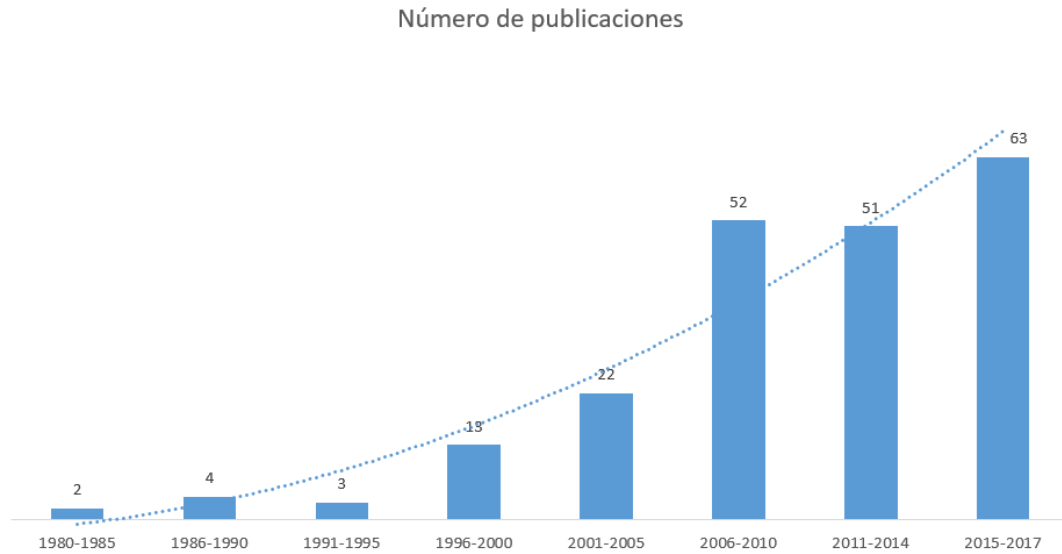


Figura 1.1: Distribución de artículos publicados por año para el MDVRP.

En esta investigación se describen algunas de las metodologías planteadas, las cuales son fundamentales para los propósitos de este trabajo. Los modelos del VRP generalmente son de tipo lineal entero mixto (MILP), en donde la complejidad del problema surge tanto en la utilización de variables enteras como a la eliminación de restricciones asociadas a problemas de subtours. Este tipo de problemas son considerados en la literatura como NP-hard y se ha recurrido a técnicas exactas, heurísticas y metaheurísticas para su solución.

Referente al problema de múltiples depósitos (MDVRP) y técnicas exactas, en el año 1985 aparece primero Kulkarni y Bhavne [3], quienes presentan una formulación lineal entera para el problema. Laporte, Nobert y Taillefer [4], en el año 1989 realizan comentarios y adecuación del modelo propuesto por Kulkarni para el MDVRP. Posteriormente, Carpaneto, Dell'amico, Fischetti y Toth [5], presentan una metodología de solución para un problema

lineal entero conocida como el Branch and Bound utilizando relajación. Estos tres autores son considerados en la literatura como los pioneros en métodos exactos aplicados al problema.

Posteriormente, debido a la complejidad del problema y la necesidad de estudiar instancias de gran tamaño, se empiezan a estudiar las técnicas metaheurísticas para resolver el MDVRP mono-objetivo. La primera metaheurística fue propuesta en el año 1996 por Renaud Laporte [6] quien estudia el MDVRP con las restricciones de capacidad de vehículos y máxima duración de las rutas (el tiempo de la ruta no puede exceder el máximo tiempo de trabajo del vehículo), el objetivo a optimizar son los costos operacionales totales, proponiendo un algoritmo de búsqueda Tabú para solucionar las instancias clásicas presentadas por Christofides y Eilon [7], Gillett y Johnson [8]. En 1999, Tüzün y Burke [9] presentan el algoritmo búsqueda Tabú para minimizar los costos totales de la ruta; además, Cordeau, Laporte y Mercier [10] en el año 2001 proponen el mismo algoritmo minimizando el número de vehículos a utilizar y con ventanas de tiempo. El primer algoritmo genético fue propuesto por Filipec, Skrllec y Krajcar [11] en el año 1997 con el objetivo de minimizar la distancia total recorrida. Vianna [12], propone un algoritmo evolutivo acoplado con la heurística de búsqueda local aplicado a la minimización de costos operacionales en 1999. A comienzos del año 2001, Thangiah y Salhi [13] presentan un algoritmo genético que define Clúster de clientes y lo resuelven utilizando una heurística de inserción, esta metodología es conocida como GenClust.

En algunos casos se encuentran técnicas metaheurísticas combinadas con heurísticas para encontrar soluciones iniciales con técnicas exactas conocidas como Matheurísticas. Lee et al. [14] plantean el MDVRP como un problema de programación dinámica determinística, con estados finitos, en una segunda etapa usan el algoritmo heurístico de la ruta más corta para su solución. Crevier et al. [15] en el año 2007 proponen una heurística que combina el método de búsqueda tabú con la programación lineal entera, en esta aproximación los vehículos pueden ser reabastecidos en depósitos intermedios a lo largo de la ruta. Además, Jeon et al. [16] proponen un algoritmo genético híbrido para el MDVRP, el cual considera el mejoramiento de una solución inicial mediante tres heurísticas y una tasa de mutación flotante para escapar de óptimos locales. Un desarrollo de dos algoritmos genéticos híbridos (HGA1 y HGA2) es presentado por Ho et al. [17] en el año 2008. La mayor diferencia entre

los genéticos es la forma en que se generan las soluciones iniciales; en el HGA1 se generan de forma aleatoria. En cambio, en el HGA2 se generan mediante el algoritmo de vecino más cercano. El desempeño de ambos algoritmos son evaluados usando dos instancias generadas de forma aleatoria donde se consideran 2 depósitos , 50 y 100 clientes.

Surekha & Sumathi [32] resuelven el MDVRP a través de un algoritmo genético. En este algoritmo los clientes se agrupan con base en la distancia al depósito más cercano y luego se realiza el ruteo usando el algoritmo de ahorros, las rutas son secuenciadas y optimizadas usando el algoritmo genético. La metodología es verificada con cinco instancias tomadas de Courdeau et al. [33]. Para el año 2009, Baldacci & Mingozzi [34], proponen una estrategia basada en conceptos de acotamiento, relajación del problema lineal y relajación lagrangeana aplicada a la formulación matemática del problema. La metodología se valida resolviendo instancias tomadas de Cordeau Gendreau, & G Laporte [33] considerando hasta 160 clientes y 5 depósitos.

En 2012 Vidal et al. [35], plantean una metodología aplicable a diferentes variantes de VRP usando para esto programación dinámica para evaluar de forma eficiente los vecindarios, combinando una secuencia basada en movimientos con una elección óptima de vehículos y depósitos. Estos conceptos se usan en dos metaheurísticas, un algoritmo de búsqueda local iterada (del inglés, Iterated Local Search -ILS) y un algoritmo genético híbrido. Con esta metodología se superan todas las respuestas de la literatura reportadas hasta la fecha.

Contardo & Martinelli [36], presentan un algoritmo exacto bajo restricciones de capacidad y longitud de ruta en el año 2014. El MDVRP es formulado usando el flujo vehicular y una formulación de conjuntos (set-partitioning formulation) los cuales son explotados en diferentes etapas del algoritmo. Validan la metodología propuesta con los casos propuestos por Cordeau Gendreau, & G Laporte [33]. Escobar et al. [42], proponen un algoritmo de búsqueda tabú granular, que considera diferentes estrategias de diversificación mejorando algunas de las soluciones, con bajos tiempo de cómputo en el año 2014.

Xu & Xia en [38], proponen un nuevo algoritmo de agrupamiento genético basado la forma geométrica en términos de la distancias recorridas por los vehículos y su capacidad. Mas tarde en 2016, Bernardes et al. en [39] presentan un algoritmo evolutivo cooperativo para dar solución al problema, mediante una descomposición en subproblemas con un único

depósito (VRP), cada parte evoluciona por separado, por lo tanto, se propone un entorno de evolución paralela con un genotipo de longitud variable y operadores de búsqueda local. En el mismo año, Tlili et al. [40], proponen un nuevo algoritmo ILS para dar solución al MDVRP. En el año 2017, Sardykov et al. [41] presentan un algoritmo branch-cut-and-price para dar solución al modelo exacto del MDVRP, encontrando óptimos globales en algunas instancias en menos de 2 horas. Toro et al. en 2018 [42], presentan un modelo matemático para dar solución al MDVRP que contempla situaciones con y sin restricción de distancia, además, se aceptan soluciones infactibles por sobrecarga en vehículos, depósitos y longitud de ruta, las cuales son llevadas como penalidades en la función objetivo. En los últimos años, nuevas variantes se han desarrollado en base al MDVRP, como el de transporte de material peligroso (HMDVRP), clientes de retorno (MDVRPB), de flota abierta que no necesita volver al depósito (MDOVRP), considerando el impacto ambiental (MDGVRP), presentados en [43, 44, 45, 46], respectivamente.

El análisis de clústeres estudia la división de objetos en grupos basado en una o varias características. Un aspecto importante de la noción de grupo, de acuerdo con Everitt [48] en el año 1974, es descrita como: regiones conectadas en un espacio multidimensional que contiene una densidad relativamente alta de puntos, separada de otras regiones de características semejantes por una que contiene una densidad relativamente baja de puntos. Jain & Dubes [47] lo describen como una técnica potencial en “dividir y conquistar” para reducir la complejidad computacional de los algoritmos de toma de decisiones en metodologías de inteligencia artificial. Esta definición de grupo es una excelente razón para usar el análisis de clústeres en la solución del MDVRPs. El potencial de las técnicas de análisis de clústeres para la solución de los problemas de ruteo VRP han sido reconocidos por autores como Dantzig & Ramser [1], Barreto et al. [23] y Zare Mehrjerdi, & Nadizadeh, [49]. En investigación previa en el año 2016 presentada por Toro, Domínguez & Escobar [50], los autores examinaron los métodos aglomerativos: Distancia Mínima, Distancia Máxima, Distancia Promedio no Ponderado, Media Aritmética, Centroide Ponderado, método de centro de sin peso, método de Ward. De acuerdo a ese se concluye que, la heurística de mejor desempeño y usada en la identificación de soluciones iniciales en el MDVRP, es la de Centroide Ponderado.

1.4. Principales resultados

Los principales resultados y contribuciones de este trabajo de grado son los siguientes:

- Comparación de diferentes técnicas y estrategias de Clústerización para generar soluciones de inicio y su efecto en el desempeño del ILS aplicada en la solución del MDVRP.
- Desarrollo del algoritmo ILS para la solución del MDVRP, implementado con base en los conceptos de intensificación y diversificación, el primero con el uso de la heurística VNS (Variable Neighborhood Search) y el segundo usando perturbación. En los dos casos, se utilizan operadores intra e inter ruta, además el algoritmo considera restricción de la longitud de la ruta. Consideran soluciones infactibles por capacidad de vehículos, depósitos y longitud de rutas, esta última en instancias con esta restricción. Las infactibilidades se suman en la función objetivo mediante el uso de factores de penalidad.

1.5. Estructura del trabajo de grado

El proyecto de grado está organizado de la siguiente forma. En el capítulo 2 se presenta el modelo matemático del MDVRP. En el capítulo 3 se describe la metodología de solución la cual incluye técnicas de clusterización, método ILS, operadores para conformar la estructura de vecindad, metodología de solución, procedimientos de clusterización y algoritmo de búsqueda local iterada (ILS). En el capítulo 4 se efectúa el análisis de resultados, finalmente en el capítulo 5 se presentan las conclusiones y trabajos futuros.

Capítulo 2

Ruteo de Vehículos con Múltiples Depósitos (MDVRP)

En el gerenciamiento óptimo de la cadena de suministros debe gestionarse de forma eficiente, el reparto de materias primas y mercancías a través de la red de distribución. Esta eficiencia se mide a menudo en las entregas oportunas a los usuarios finales. El problema de ruteo considerando múltiples depósitos (MDVRP) es formulado como una red logística, que pretende resolver, entre otros aspectos, la atención a los clientes en el menor tiempo posible.

El MDVRP es una generalización del CVRP (Capacited Vehicle Routing Problem), considerando más de un depósito, donde cada vehículo debe iniciar y terminar el recorrido en el mismo depósito. La figura 2.1, presenta la numeración de los depósitos con su respectiva capacidad (Q), así mismo para los clientes la demanda (d). Las aplicaciones del MDVRP surgen cuando una empresa tiene varios almacenes cada uno de ellos con su propia flota de vehículos, que en conjunto son capaces de satisfacer la demanda de los clientes [18].

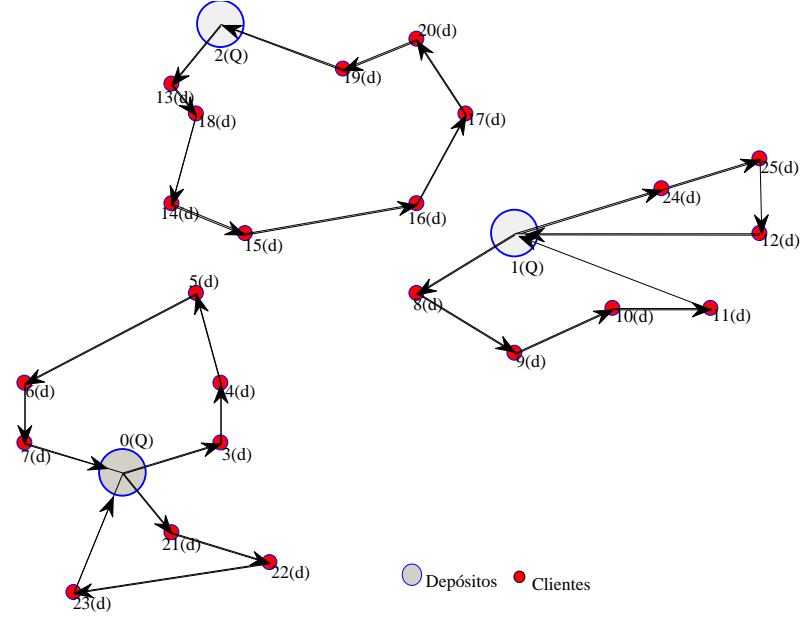


Figura 2.1: MDVRP para 23 clientes y 3 depósitos

2.1. Marco conceptual

Para el desarrollo de este proyecto se mencionan algunos conceptos básicos relacionados con los problemas característicos en la distribución y transporte, los cuales consisten básicamente en asignar una ruta a cada vehículo de una flota cuyo fin es la distribución de la carga. Según H.K.Chen, et al. en [19] se pueden clasificar de la siguiente manera.

- Por el tamaño de la flota: Un solo vehículo, el número de vehículos limitado, o el número de vehículos ilimitado.
- Por el tipo de flota: un solo tipo de vehículo u homogénea, con múltiples tipos de vehículos o heterogénea, y con o sin compartimientos para las mercancías.
- Depósito o almacén. Único, múltiples depósitos y depósitos de reaprovisionamiento intermedio.
- Por naturaleza de la demanda: Determinística o estocástica, dinámica.

- Localización de la demanda: En los nodos, arcos o mixta.
- Restricciones horarias de servicio: Sin restricciones, en el inicio o en el final de la ruta, y de ventanas temporales únicas o múltiples.
- Ciclos de servicio: Único ciclo o varios recorridos.
- Restricciones en la capacidad de los vehículos: Limitada y única para todos o limitada y diferente para algunos vehículos.
- Velocidad media de los vehículos: velocidad constante, dependiente del horario y/o dependiente del recorrido.
- Duración máxima de una ruta: limitada e igual para todas, diferente, estricta o flexible.
- Número de rutas por vehículo: Única o varias rutas por vehículo.
- Costo: El costo puede ser fijo, variable o mixto, también se puede considerar penalizaciones por demandas insatisfechas.
- Función Objetivo: Minimizar costos fijos o variables en el recorrido, minimización de rutas o vehículos a usar, minimizar tiempo empleado, minimizar esperas, maximizar beneficio, maximizar función de utilidad del proveedor, entre otros.

El MDVRP se basa principalmente en una extensión del VRP, consistiendo en que el total de la demanda de las ciudades visitadas en una ruta no debe superar la capacidad del vehículo. El objetivo puede ser reducir al mínimo la suma de la distancia recorrida por todas las rutas, el mínimo número de vehículos, o una combinación de ambos criterios. Cabe notar que la dirección de la ruta no es una restricción al problema.

Los problemas de programación entera y enteros mixtos como el MDVRP son difíciles de resolver, esto es debido a que el número de alternativas de solución incrementa exponencialmente de acuerdo al tamaño del problema. Así que, esto hace que los problemas grandes de programación entera sean extremadamente difíciles de resolver usando algoritmos de matemáticas exactas existentes, teniendo que recurrir a técnicas metaheurísticas para su solución.

Un ejemplo de la complejidad de los problemas, es el TSP (agente viajero), donde un mensajero tiene que visitar n clientes. En la Tabla 2.1 se muestra como el tiempo computacional (en segundos) se incrementa de acuerdo al tamaño del problema, si se decide enumerar de manera exhaustiva el total de alternativas posibles.

Tabla 2.1: Incremento computacional del problema TSP.

Tamaño n	Número de alternativas $(n - 1)!$
3	2
5	24
9	40320
13	479001600
17	20922789000
20	121645100408832000

Las soluciones comunes para resolver problemas enteros (PE) son: redondear la solución no entera, branch-and-bound, cutting plane, branch-and-cut, branch-and-price.

Redondear la aproximación consiste en emplear el método simplex y relajar las condiciones enteras. Sin embargo, redondear es un proceso que no garantiza optimalidad y las soluciones obtenidas, como resultado del procedimiento, podrían ser infactibles [20].

2.2. Modelo Matemático

La representación matemática del MDVRP se construye a partir del modelo del CVRP, al cual se le adiciona la consideración de más de un depósito. Como objetivo principal está la minimización del costo de recorrido de los vehículos.

Sea $G = (V, E)$ un grafo orientado, donde V es el conjunto de nodos descrito y E es el conjunto de arcos o aristas conectando cada par de nodos. El conjunto V es posteriormente particionado en dos subconjuntos: $V_C = \{v_1, v_2, \dots, v_N\}$ conjunto de clientes a ser atendidos, y $V_d = \{v_{N+1}, v_{N+2}, \dots, v_M\}$ conjunto de depósitos. Cada cliente $v_i \in V_c$ tiene una demanda no negativa d_i . Cada arco pertenece al conjunto E que tiene asociado un costo, distancia

o tiempo de viaje c_{ij} . Hay un total de N vehículos, cada uno con una capacidad Q_k . El problema consiste en determinar un conjunto de vehículos de tal manera que: 1) la ruta de cada vehículo debe iniciar y finalizar en el mismo depósito; 2) cada cliente es servido exactamente una vez por un vehículo; 3) el total de la demanda de cada ruta no debe exceder la capacidad del vehículo.

2.2.1. Nomenclatura

La nomenclatura para los conjuntos, parámetros y variables de decisión para el MDVRP se describe a continuación.

Conjuntos

I	Conjunto de depósitos $I \in V$
J	Conjunto de clientes $J \in V$
K	Conjunto de vehículos $K \in V$

Parámetros

N	Número de vehículos
c_{ij}	Distancia entre los nodos i y j .
W_i	Capacidad del depósito
d_i	Demanda del cliente i
Q_k	Capacidad del vehículo (ruta) k
C_{ij}	Costo asociado a ir del nodo i al nodo j .

Variables de decisión

x_{ijk}	Variable binaria que indica que el nodo i precede al nodo j en la ruta k .
Z_{ij}	Variable binaria que define si el consumidor ubicado en el nodo j es atendido por el centro de distribución i .
Ulk	Variable auxiliar usada en las restricciones de eliminación de sub-toures en la ruta k .

2.2.2. Estructura matemática del modelo

$$\min = \sum_{i \in I \cup J} \sum_{j \in I \cup J} \sum_{k \in K} C_{ij} * x_{ijk} \quad (2.1)$$

s.a

$$\sum_{k \in K} \sum_{i \in I \cup J} x_{ijk} = 1 \quad j \in J \quad (2.2)$$

$$\sum_{j \in J} d_j \sum_{i \in I \cup J} x_{ijk} \leq Q_k \quad k \in K \quad (2.3)$$

$$U_{lk} - U_{jk} + N x_{ijk} \leq N - 1 \quad l, j \in J, k \in K \quad (2.4)$$

$$\sum_{j \in I \cup J} x_{ijk} - \sum_{j \in I \cup J} x_{jik} = 0 \quad k \in K, i \in I \cup J \quad (2.5)$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1 \quad k \in K \quad (2.6)$$

$$\sum_{j \in J} d_j Z_{ij} \leq W_i \quad i \in I \quad (2.7)$$

$$-z_{ij} + \sum_{u \in I \cup J} (x_{iuk} + x_{ujk}) \leq 1 \quad i \in I, j \in J, k \in K \quad (2.8)$$

$$x_{ijk} \in \{0, 1\} \quad i \in I, j \in J, k \in K \quad (2.9)$$

$$Z_{ij} \in \{0, 1\} \quad i \in I, j \in J \quad (2.10)$$

$$U_{lk} \geq 0 \quad l \in J, k \in K \quad (2.11)$$

La función objetivo (2.1) minimiza la distancia total de todas las rutas. La ecuación (2.2) establece que cada cliente debe ser asignado a una ruta. La ecuación (2.3) representa la restricción de capacidad del conjunto de vehículos. El conjunto de restricciones de eliminación de sub-tours está representado por (2.4). Las restricciones de conservación de flujo son expresadas mediante (2.5). La ecuación (2.6) garantiza que cada vehículo atienda una única ruta. Las restricciones de capacidad de los depósitos está dada por (2.7). La ecuación (2.8) especifica que un cliente puede ser asignado a un depósito, únicamente si hay una ruta que

parte desde el depósito y transita a través del cliente. Las restricciones (2.9), (2.10) definen la naturaleza binaria de las variables $x_{ijk}, Z_{i,j,k}$. Finalmente la restricción (2.11) define U_{lk} como una variable no negativa continua.

Capítulo 3

Método de solución

En esta investigación se utiliza una metodología en dos etapas, en la primera se determina una solución inicial mediante **Técnicas Aglomerativas de Clústerización**, en la segunda se aplica el algoritmo **ILS** el cual a partir de la solución obtenida con las técnicas de clusterización, inicia una etapa de mejoramiento y son descritas a seguir.

3.1. Técnicas de Clústerización

El análisis de conglomerados o clústers [28], estudia la división de los espacios como objetos o individuos en grupos basados en una o varias de sus características. Donde la definición de grupos, de acuerdo con Jain y Dubes [30], puede ser descrito como regiones conectadas de un espacio multidimensional que contiene una densidad relativamente alta de puntos semejantes, separada de otras regiones por una región que contiene una densidad relativamente baja de puntos con diferentes características. Esta definición de grupo, puede ser utilizada en el análisis de clúster y en la resolución de los problemas de ruteo, por ejemplo, el reconocimiento de grupos de clientes (ó ciudades) puede ser un buen comienzo para obtener buenas soluciones, ya que parece natural que una conexión relacionada con dichas ciudades tenderá a construir rutas que abarcan regiones de alta densidad (aglomeraciones de población).

Por lo tanto, en muchos casos se pueden emitir juicios a priori de que los clientes en ciertas regiones geográficas no recibirán servicios de posibles almacenes en otras regiones, además,

los clientes pueden agruparse frecuentemente en concentraciones de demanda (por ejemplo, supermercados de cadena y almacenes mayoristas) debido a la proximidad geográfica. Siendo así, los conocidos métodos jerárquicos tienen por objetivo agrupar clústers para formar uno nuevo o bien separar alguno ya existente para dar origen a otros dos, de tal forma que, si sucesivamente se va efectuando este proceso de aglomeración o división, se minimice alguna distancia o bien se maximice alguna medida de similitud. Los métodos jerárquicos se subdividen en aglomerativos y disociativos. Cada una de estas categorías presenta una gran diversidad de variantes y descritos de forma resumida así:

- Los métodos aglomerativos, también conocidos como ascendentes, comienzan el análisis con tantos grupos como individuos haya. A partir de estas unidades iniciales se van formando grupos, de forma ascendente, hasta que al final del proceso todos los casos tratados están englobados en un mismo conglomerado. Estos métodos son utilizados en esta investigación.
- Los métodos disociativos, también llamados descendentes, constituyen el proceso inverso al anterior. Comienzan con un conglomerado que engloba a todos los casos tratados y, a partir de este grupo inicial, a través de sucesivas divisiones, se van formando grupos cada vez más pequeños. Al final del proceso se tienen tantas agrupaciones como casos han sido tratados.

En las técnicas aglomerativas, sea n el conjunto de la muestra, de donde resulta el nivel $K=0$, con n grupos. En el siguiente nivel se agruparán aquellos dos individuos que tengan la mayor similitud (o menor distancia), resultando así $n - 1$ grupos; a continuación y siguiendo con la misma estrategia se agruparán en el nivel posterior, aquellos dos individuos (o clústeres ya formados) con menor distancia o mayor similitud; de esta forma, en el nivel L se tendrá $n - L$ clústeres formados. Si se continua agrupando de esta forma, se llega al nivel $L = n - 1$ en el que sólo hay un grupo, formado por todos los individuos de la muestra.

Los métodos jerárquicos permiten la construcción de un árbol de clasificación, que recibe el nombre de dendrograma (ver Figura 3.1), en el cual se puede seguir de forma gráfica el procedimiento de unión seguido, mostrando que grupos se van uniendo, en que nivel concreto lo hacen, así como el valor de la medida de asociación entre los grupos cuando estos se agrupan

(valor conocido como nivel de fusión). En resumen, la forma general de operar los métodos aglomerativos parte de tantos grupos como individuos haya, posteriormente se selecciona una medida de similitud agrupándose los 2 clústeres con mayor similitud, así se continúa hasta que:

- Se forma un sólo grupo
- Se alcanza el número de clústeres prefijado
- Se detecta que hay razones estadísticas para no continuar agrupando clústeres, ya que los más similares no son lo suficiente homogéneos como para determinar una misma agrupación.

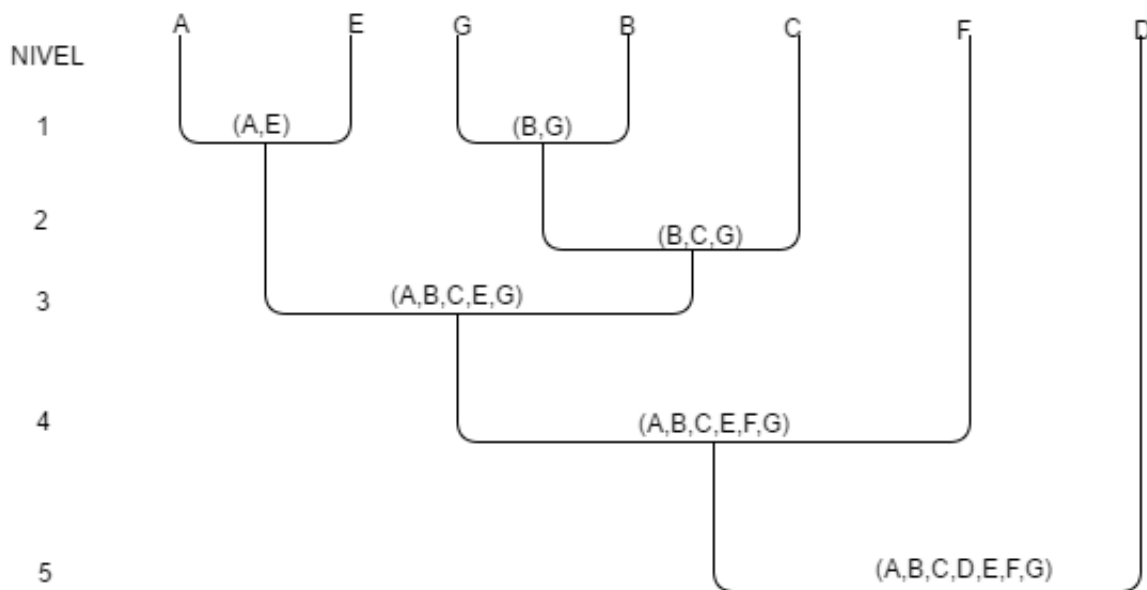


Figura 3.1: Dendrograma (árbol de clasificación)

Los Métodos Jerárquicos y otros algoritmos de agrupamiento representan un intento para encontrar "buenos" clústeres de una nube de datos. Habitualmente, se hace imposible evaluar todas las posibilidades de agrupamiento, cuando los conjuntos de datos son de gran volumen [21]. Varios autores han utilizado técnicas de agrupamiento en diferentes algoritmos para

resolver diferentes variantes del VRP entre ellos, Giosa et al.[22] en el MDVRP y Barreto et al. [23] en el CLRP (*Green Capacitated Location Routing Problem*), ambos con resultados eficientes. Los algoritmos de agrupamiento jerárquico implican un proceso secuencial, quiere decir, es necesario aplicar cada técnica de manera consecutiva, y en cada paso del proceso jerárquico de agrupamiento, una observación o un conjunto de observaciones se va fusionando en otro grupo. En todos los métodos se inicia el procedimiento con base en la matriz de distancias entre clientes, definida como la distancia inicial entre cada clúster, además, es necesario tener en cuenta que al inicio del proceso cada cliente forma un clúster. C_i representa el clúster i con n_i clientes y C_j representa el clúster j con n_j clientes.

Las diferentes métricas de Clústerización usadas en este estudio son: Distancia Mínima, *Single Linkage (SL)*; Distancia Máxima, *Complete Linkage (CL)*; Distancia Promedio no Ponderada, *Unweight Aritmethic Average (UAL)*, Promedio Ponderado, *Weighted Aritmethic (AL)*, Método del Centroide Ponderado, *Weighted Centroid Method (WCM)*, Método de la Mediana, *Centroid Method Unweight (CMU)*, Método de Ward, *Ward's Method (W)*. [21].

El análisis de conglomerados considera diferentes métodos de agrupamiento, según el tipo de variables (cuantitativo, cualitativo, binario, o mixto). En esta investigación se utilizan las medidas cuantitativas y se proponen varias de éstas para determinar la proximidad entre puntos del plano, sin embargo la métrica cuantitativa más común es la Euclidiana, que determina la proximidad entre los puntos $I = (X_i, Y_i)$ y $J = (X_j, Y_j)$.

A continuación, se presentan estrategias que pueden ser empleadas en el agrupamiento de Clústeres.

3.1.1. Distancia Mínima (SL)

En el método de Distancia Mínima, la distancia entre dos Clústeres C_i y C_j se define como la mínima entre un par de puntos de acuerdo con 3.1. En cada paso se determina la distancia para cada par de Clústeres y se fusionan los dos Clústeres con menor distancia. El número de Clústeres se reduce en cada paso, al final del procedimiento se tiene un sólo clúster.

$$d(C_i, C_j) = \min_{\substack{x_l \in C_i \\ x_m \in C_j}} d(x_l, x_m); l = 1, \dots, n_i, \quad m = 1, \dots, n_j \quad (3.1)$$

3.1.2. Distancia Máxima (CL)

La distancia entre C_i y C_j es definida como la máxima distancia entre un punto (cliente) en C_i y un punto (cliente) en C_j . En cada paso, la distancia expresada en 3.2 es encontrada para cada par de Clústeres, y los 2 Clúster con la distancia de separación más grande son fusionados.

$$d(C_i, C_j) = \max_{\substack{x_l \in C_i \\ x_m \in C_j}} d(x_l, x_m); l = 1, \dots, n_i, \quad m = 1, \dots, n_j \quad (3.2)$$

3.1.3. Distancia Promedio no Ponderada (UAL)

Este método toma la medida de la distancia entre un par de Clústeres C_i (con n_i clientes), que a su vez consta de 2 grupos C_{i1} y C_{i2} (con n_{i1} y n_{i2} clientes respectivamente) y C_j (n_j clientes). La distancia es calculada por 3.3.

$$d(C_i, C_j) = \frac{d(C_{i1}, C_j) + d(C_{i2}, C_j)}{2} \quad (3.3)$$

3.1.4. Distancia Promedio Ponderada(AL)

En este método es considerado que la distancia entre dos Clústeres es definida por el peso promedio de la distancia de los componentes Clústerizados respecto al otro. Quiere decir, se tienen dos Clústeres C_i y C_j . El Clúster C_i está conformado por otros 2 Clústeres C_{i1} y C_{i2} con n_{i1} y n_{i2} número de elementos (clientes) respectivamente. Sea $n_i = n_{i1} + n_{i2}$ el número de elementos, C_i y n_j el número de elementos que compone a C_j . La distancia es calculada por 3.4.

$$d(C_i, C_j) = \frac{n_{i1}d(C_{i1}, C_j) + n_{i2}d(C_{i2}, C_j)}{n_i} \quad (3.4)$$

3.1.5. Método del Centroide Ponderado (WCM)

En este método es usada la distancia Euclidiana y distancia Euclidiana cuadrática, no importa cual de las dos se use, el procedimiento es el mismo. Este método esta basado en

la similitud en cada centro de los Clústeres. Se define la distancia de dos clúster como se muestra a continuación:

Sea dos Clústeres C_i y C_j , se supone que el clúster C_i está formado, en principio, por dos Clústeres, C_{i1} y C_{i2} con n_{i1} y n_{i2} elementos (ciudades) respectivamente, por lo tanto, sea $n_i = n_{i1} + n_{i2}$ el número de elementos que conforman C_i y n_j los elementos que forman C_j . La distancia es calculada por 3.5.

$$d(C_i, C_j) = \frac{n_{i1}d^2(C_{i1}, C_j)}{n_{i1} + n_{i2}} + \frac{n_{i2}d^2(C_{i2}, C_j)}{n_{i1} + n_{i2}} - \frac{n_{i1}n_{i2}d^2(C_{i2}, C_j)}{(n_{i1} + n_{i2})^2} \quad (3.5)$$

3.1.6. Método de la Mediana (CMU)

El método puede ser utilizado con distancia euclidiana o distancia euclidiana cuadrática, no importa con cual se use, el proceso es semejante. Este método se basa en la similaridad de los centros de cada Clúster, la principal diferencia con el método del centroide ponderado es que considera que $n_{i1} = n_{i2}$, haciendo que el centroide del Clúster C_i se localice entre los clúster C_{i1} y C_{i2} , por lo tanto, el centroide del Clúster (C_i, C_j) esta localizado en el punto medio o mediana del triangulo formado por C_{i1} , C_{i2} y C_j Clústeres. La distancia entre el par de Clústeres es calculada por 3.6.

$$d(C_i, C_j) = \frac{d^2(C_{i1}, C_j)}{2} + \frac{d^2(C_{i2}, C_j)}{2} - \frac{d^2(C_{i2}, C_j)}{4} \quad (3.6)$$

3.1.7. Método de Ward (W)

En este método para cada etapa, se unen los dos Clústeres para los cuales el aumento más pequeño se toma en el valor total de la suma de los cuadrados de las diferencias dentro de cada Clúster. Cada individuo en el centroide del Clústeres, tiene como objetivo minimizar (3.7).

Donde:

E_k :Es la suma de los errores medios cuadráticos de cada Clúster k (distancia euclidiana cuadrática entre cada individuo del Clúster a su centroide).

$(x_{ij}^k - m_j^k)^2$: Representa la distancia Euclidiana cuadrática de un cliente perteneciente a el

Clúster K, donde el primer elemento corresponde a un vector de coordenadas que describe el cliente perteneciente al Clúster K y el segundo elemento corresponde al centroide del Clúster K.

$$\min E_k = \sum_{i \in n_k} \sum_{j \in n} (x_{ij}^k - m_j^k)^2 \quad (3.7)$$

Tabla 3.1: Métricas de Clústerización

Método	Métrica de distancia
SL	$d(C_i, C_j) = \min_{\substack{x_l \in C_i \\ x_m \in C_j}} d(x_l, x_m); l = 1, \dots, n_i, \quad m = 1, \dots, n_j$
CL	$d(C_i, C_j) = \max_{\substack{x_l \in C_i \\ x_m \in C_j}} d(x_l, x_m); l = 1, \dots, n_i, \quad m = 1, \dots, n_j$
UAL	$d(C_i, C_j) = \frac{d(C_{i1}, C_j) + d(C_{i2}, C_j)}{2}$
AL	$d(C_i, C_j) = \frac{n_{i1}d(C_{i1}, C_j) + n_{i2}d(C_{i2}, C_j)}{2}$
WCM	$d(C_i, C_j) = \frac{n_{i1}d^2(C_{i1}, C_j)}{n_{i1} + n_{i2}} + \frac{n_{i2}d^2(C_{i2}, C_j)}{n_{i1} + n_{i2}} - \frac{n_{i1}n_{i2}d^2(C_{i2}, C_j)}{(n_{i1} + n_{i2})^2}$
CMU	$d(C_i, C_j) = \frac{d^2(C_{i1}, C_j)}{2} + \frac{d^2(C_{i2}, C_j)}{2} - \frac{d^2(C_{i2}, C_j)}{4}$
W	$\min E_k = \sum_{i \in n_k} \sum_{j \in n} (x_{ij}^k - m_j^k)^2$

Algunos autores han intentado demostrar la superioridad de una u otra medida de proximidad. A pesar de estos esfuerzos, todavía no se tiene una idea clara sobre las potencialidades generales o específicas de cada uno. La evolución de los diferentes estudios, refuerza la idea de que no existe una medida adecuada para todas las aplicaciones y casos. En cada caso, la medida debe elegirse cuidadosamente y, probablemente, sólo después de las pruebas con varias medidas, se pueden extraer conclusiones definitivas [29]. La Tabla 3.1 presenta las ecuaciones de las métricas de manera resumida que serán utilizadas en la metodología propuesta.

3.2. Búsqueda Local Iterada (ILS)

La técnica metaheurística búsqueda Local Iterada o *Iterated Local Search* "de sus siglas en Inglés (ILS)" ([24],[25]) basa su funcionamiento en la aplicación reiterada de perturbaciones y búsquedas locales estocásticas sobre una solución dada. Fue mencionada por primera vez en la tesis doctoral de T.Stützle [26]. El algoritmo ILS tiene como entrada una solución inicial generada por una heurística, generalmente es tratado como siendo de tipo trayectoria y se implementa con base a dos procedimientos denominados genéricamente como diversificación e intensificación (ver figura 3.2). En la primera etapa, *Proceso de Intensificación*, el algoritmo aplica una operación de búsqueda local (LS-Local Search) sobre la solución actual. Seguidamente, se realiza el *Proceso de Diversificación* con el fin de iniciar la búsqueda en otra parte del espacio solución, también se aplica un criterio de aceptación que determina la nueva solución generada a partir de la operación del ILS que sustituye la solución previa. En muchos casos, esta operación únicamente consiste en comparar la calidad de ambas soluciones, conservando aquella que proporciona una mejor solución al problema, sin embargo, en otras el esquema básico de la metaheurística se mejora añadiendo alguna característica adicional, como puede ser la utilización de un histórico que determine el reinicio del algoritmo si tras un determinado número de iteraciones no se consigue mejorar la solución actual (salir de óptimos locales).

Dado el espacio de soluciones SS , una vecindad N y un procedimiento de búsqueda local ILS, se establece una correspondencia entre SS y el espacio de soluciones de óptimos locales SS^* . ILS se basa en un procedimiento de búsqueda local sobre el espacio de soluciones de óptimos locales SS^* y perturbar la vecindad de soluciones N a una nueva vecindad N^* .

El proceso de intensificación es resuelto mediante la heurística de búsqueda en vecindario variable, o VNS (Variable Neighborhood Search) y la segunda usando operadores inter-rutas, empleadas como heurísticas de perturbación. La Figura 3.2 ilustra el movimiento de la solución en la aplicación del algoritmo ILS. En el VNS se usan operadores inter e intra-ruta, con los cuales se generan vecindarios alrededor de un punto, el proceso termina al no obtenerse soluciones de mejor calidad, el criterio de parada lo determina los operadores inter-rutas, al no ser identificadas soluciones de mejor calidad. La perturbación consiste en aplicar 2 operadores inter-ruta de manera consecutiva, que permite que el procedimiento logre salir de óptimos

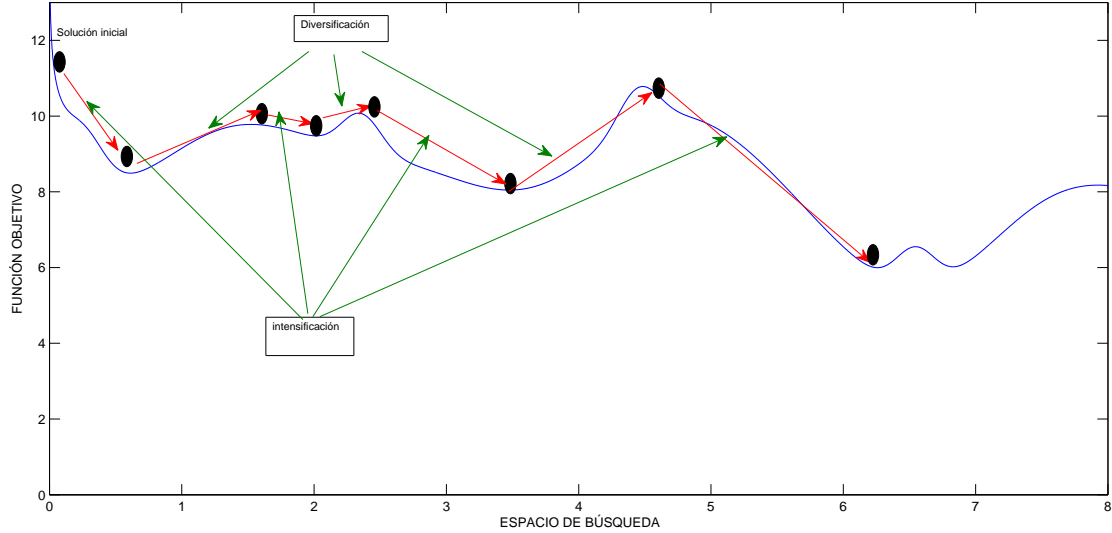


Figura 3.2: Proceso de intensificación y diversificación

locales y alcanzar otras regiones del espacio de búsqueda.

Cada movimiento de la estructura de vecindad es cuantificado con la variación de costos ΔC (3.8) y de las infactibilidades en la carga en vehículos, los depósitos y longitud de ruta en instancia con este tipo de restricción. El operador de vecindad seleccionado actualiza la función de adaptación, costos e infactibilidades por exceso de carga en vehículos y depósitos, como también infactibilidades por exceso en las longitudes de rutas, en instancias con esta restricción.

$$\Delta C = C^0 - C^1 \quad (3.8)$$

En el ILS, la operación de más relevancia es la búsqueda local (intensificación). Esta operación debe ser tan efectiva y rápida como sea posible, y a través de su uso puede ser mejorada la solución actual, y cuanto más efectiva sea, mejores serán los resultados que arroje el algoritmo ILS. Visualizando desde un entorno de programación, la búsqueda local es una función "caja negra" que será utilizada por el algoritmo en cada iteración. El diseño de un buen algoritmo de búsqueda local, muchas veces se trata de una operación no trivial, donde en ocasiones es más importante la rapidez del algoritmo que su eficiencia. La perturbación (diversificación), debe ser complementaria en el proceso de búsqueda local, y

en muchas ocasiones resulta ser crítica por su elección y determinante en el desempeño del algoritmo. El algoritmo debe garantizar un buen equilibrio entre los procesos de intensificación y diversificación. Por otro lado, la primer solución del algoritmo (heurístico constructivo), se vuelve muy importante en el desarrollo del mismo. El procedimiento de cómputo disminuye drásticamente el tiempo computacional, cuando se usa una solución inicial de buena calidad, comparada cuando se usan soluciones aleatorias y para soluciones finales de igual calidad. Además de mejoras en el tiempo computacional, en los procedimientos que usan buenos inicializadores se obtienen soluciones de mejor calidad, respecto a aquellas que usan arranques generados aleatoriamente. El Algoritmo 1 resume el funcionamiento del ILS.

Algorithm 1 Pseudocódigo de la búsqueda local iterativa (ILS)

```
1: Requiere: Generar_Solución_inicial.  
2: Empezar: búsqueda_local (solución).  
3: while iteraciones < limite do  
4:   nueva_solucion  $\leftarrow$  Pertubar (solución)  
5:   nueva_solucion  $\leftarrow$  búsqueda_local( nueva solución)  
6:   solucion  $\leftarrow$  actualizar_solucion (solución,nueva_solución)  
7: end while  
8: return solución
```

Cómo se observa en el algoritmo, se genera una solución inicial al inicio del proceso (puede ser de manera aleatoria, línea 1). Posteriormente, se aplica el método de búsqueda local (línea 2) para realizar el proceso de intensificación y encontrar un óptimo de buena calidad. De no presentar mejora, la solución es perturbada y evita una búsqueda infructuosa (línea 4). Después el proceso de solución es mejorado mediante el procedimiento de búsqueda local (intensificación, línea 5). Finalmente, la nueva solución es comparada con la existente (línea 6), almacenando la solución de mejor calidad. Las soluciones de mala calidad son eliminadas, y el algoritmo continua su procedimiento hasta que se cumpla el máximo de iteraciones (límite).

3.3. Operadores para la construcción de esquemas de vecindad

Los operadores son procedimientos que mediante su uso es posible simular esquemas de vecindad en las etapas de diversificación e intensificación en el algoritmo ILS. Los operadores pueden ser programados de dos maneras: la primera es aquella en la cual su efecto es sentido en una sola ruta y denominado intra-ruta, la segunda es aquella en la cual su efecto es sentido por dos rutas y denominado inter-ruta. Haciendo uso de los operadores inter e intra-ruta, es posible generar soluciones en la vecindad en la etapa de intensificación y de cambio de vecindario en la etapa de diversificación y al ser usados de manera coordinada logran identificar soluciones de buena calidad. En su tesis Doctoral, Subramanian(2012) [27] presenta algunas de las estructuras de vecindario, sus procedimientos de verificación de factibilidad y algoritmos. La descripción de los operadores se presentan a continuación.

3.3.1. Shift(1,0):

Un cliente k es transferido desde la ruta r_1 a la ruta r_2 (inter-ruta).

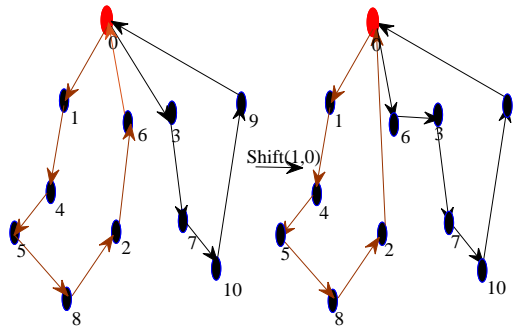


Figura 3.3: Operador Shift(1,0) inter-ruta

3.3.2. Swap(1,1):

Es un intercambio (inter-ruta) entre el cliente k de la ruta r_1 y el cliente l de la ruta r_2 .

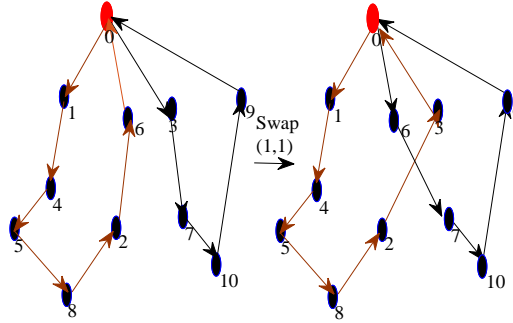


Figura 3.4: Operador Swap(1,1) inter-ruta

3.3.3. Shift(2,0):

Dos clientes k y l adyacentes (aquellos que están comunicados por una misma ruta), son transferidos de la ruta r_1 a la ruta r_2 .

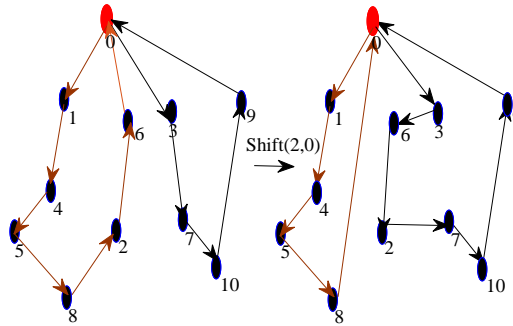


Figura 3.5: Operador Shift(2,0) inter-ruta

3.3.4. Swap(2,1):

Dos clientes k y l adyacentes (aquellos que están comunicados por una misma ruta) de la ruta r_1 son cambiados por un cliente k' de la ruta r_2 .

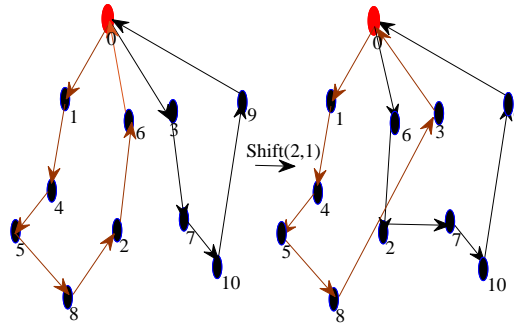


Figura 3.6: Operador Swap(2,1)inter-ruta

3.3.5. Swap(2,2):

Permutación entre dos clientes adyacentes, k y l , de la ruta r_1 por otros dos clientes adyacentes k' y l' pertenecientes a la ruta r_2 .

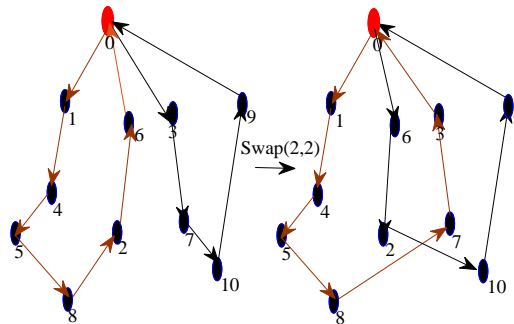


Figura 3.7: Operador Swap(2,2) inter-ruta

3.3.6. 2-OPT:

Este operador puede ser del tipo intra-ruta e inter-ruta, consiste en eliminar 2 arcos existentes y crear 2 nuevos.

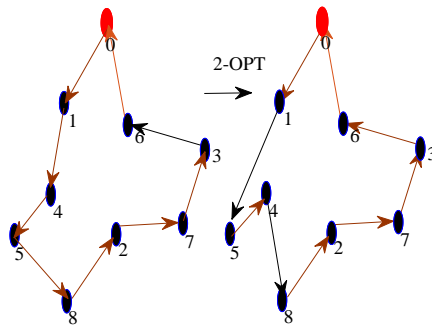


Figura 3.8: Operador 2-OPT intra-ruta

3.3.7. Split:

Este operador permite separar una ruta en 2, quedando conectado al mismo depósito, tal como se ilustra en la Figura 3.9.

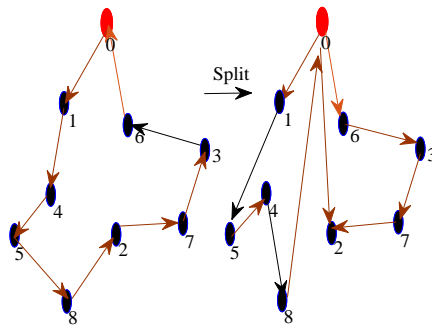


Figura 3.9: Operador Split inter-ruta

3.4. Metodología propuesta

En esta investigación se propone una metodología que combina técnicas aglomerativas de Clusterización para identificar alternativas iniciales y el algoritmo ILS en la resolución del MDVRP con el fin de dar solución a dicho problema de manera eficiente.

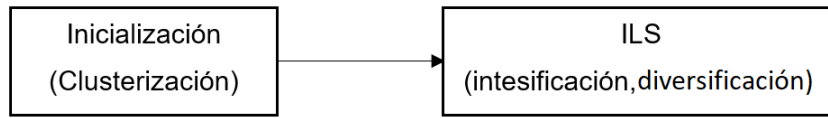


Figura 3.10: Metodología integrada en dos etapas.

En la primera etapa, se lleva a cabo un estudio que usa diferentes técnicas de clusterización, y cuyo objetivo es identificar soluciones de buena calidad que sirvan de inicializadores para el algoritmo ILS. La segunda etapa consiste en utilizar el algoritmo ILS basado en los conceptos de intensificación y diversificación, el primero se lleva a cabo mediante el uso del algoritmo VNS (del inglés, Variable Neighborhood Search), el segundo aplicando esquemas de perturbación aleatorias, en esta etapa, como en la anterior se hace uso de los operadores descritos anteriormente. En el algoritmo ILS son usados operadores intra e inter ruta en la implementación de los procedimientos de intensificación y diversificación. Una de las características del algoritmo está en que realiza la búsqueda tanto en la región factible como en la infactible y de esta forma el procedimiento es más flexible. Las infactibilidades consideran excesos en capacidad de depósitos y vehículos, como también excesos en la longitud de rutas, en instancias con restricción de distancia. Estas infactibilidades son llevadas en cuenta como penalizaciones (β) en la función objetivo (F_{obj}) a través de una función de adaptación (F_{ad}) como se muestra en la ecuación (3.9). Así el algoritmo implementado resuelve dos tipos de problemas, con y sin restricción de longitud de las rutas.

$$F_{ad} = F_{obj} + \beta \quad (3.9)$$

Para aplicar los operadores inter-ruta, se usan criterios de granularidad, que consisten en

tener en cuenta distancia máxima permitida en la que una ruta puede desviarse para atender a un cliente. Si dichas longitudes de conexión no sobrepasan un valor especificado, el cliente es aceptado para realizar transferencia. El valor de distancia máxima de conexión es calculado como la distancia promedio del depósito al que pertenece el cliente y los depósitos vecinos. Esta distancia es multiplicada por un factor seleccionado aleatoriamente entre $[0.1 - 1.0]$.

3.5. Procedimiento de Clústerización

Soluciones de buena calidad al inicio de los procesos de optimización combinatorial, permiten ahorros significativos en tiempo computacional en el proceso de las metaheurísticas. Las heurísticas constructivas identifican soluciones de buena calidad en tiempos de cómputo razonables, que sumado al tiempo de cómputo requerido por la metaheurística, resulta menor que el requerido por la misma, iniciando con soluciones generadas aleatoriamente y para soluciones finales de igual calidad. En esta sección serán usados algoritmos heurísticos constructivos para la generación de las soluciones iniciales. Los algoritmos serán generados con base en diferentes procedimientos de clusterización y haciendo uso de los operadores inter e intra-ruta. Se lleva a cabo a través de un adecuado uso de la estructura de datos. Esta estructura es implementada utilizando una disposición de árbol, donde los terminales corresponden a los Clústeres. La distancia entre ellos corresponde según la métrica elegida. En la tabla 3.2 se describe un caso de una instancia de 7 clientes, así. En el Nivel I se tienen 7 Clústeres debido a que cada individuo es un Clúster en el paso inicial, a medida que se va ascendiendo se va disminuyendo la cantidad de clústeres y sigue el siguiente procedimiento:

Tabla 3.2: Estructura tipo árbol.

Nivel	No. Clústeres	Clúster	Clúster	Clúster	Clúster	Clúster	Clúster	Clúster
1	7	A	B	C	D	E	F	G
2	6	(C,E)	A	B	D	F	G	
3	5	(D,F)	(C,E)	A	B	G		
4	4	(C,E,D,F)	A	B	G			
5	3	(A,C,E,D,F)	B	G				
6	2	(A,B,C,E,D,F)	G					

3.5.1. Procedimiento para generar la solución inicial

La metodología para generar las soluciones iniciales se representa en los diagramas de flujo de las figura 3.11 y 3.12. Este procedimiento tiene como insumo la información de Clústeres de la Tabla 3.2. El diagrama de flujo de la figura 3.11 es un procedimiento interno del diagrama de flujo de la figura 3.12 y entre ambos establecen una búsqueda exhaustiva hasta encontrar la configuración de menor costo posible. El resultado de este procedimiento corresponde a la solución inicial del algoritmo ILS.

En el diagrama de flujo de la figura 3.11 es necesario tener las siguientes consideraciones:

- Cada Clúster tiene una estructura de datos formada por: centroide, demanda, lista de los depósitos más cercanos en orden ascendente de acuerdo con la distancia euclidiana respecto al centroide.
- Punto de búsqueda inicial: es un clúster del arreglo de Clústeres compuesto por uno o varios clientes, ubicado en la fila seleccionada n y columna número uno (Tabla 3.2).
- Cuando se asigna un clúster, se asignan todos los clientes del clúster al depósito analizado.
- En el arreglo de Clústeres cada fila del arreglo tiene diferente número de columnas.

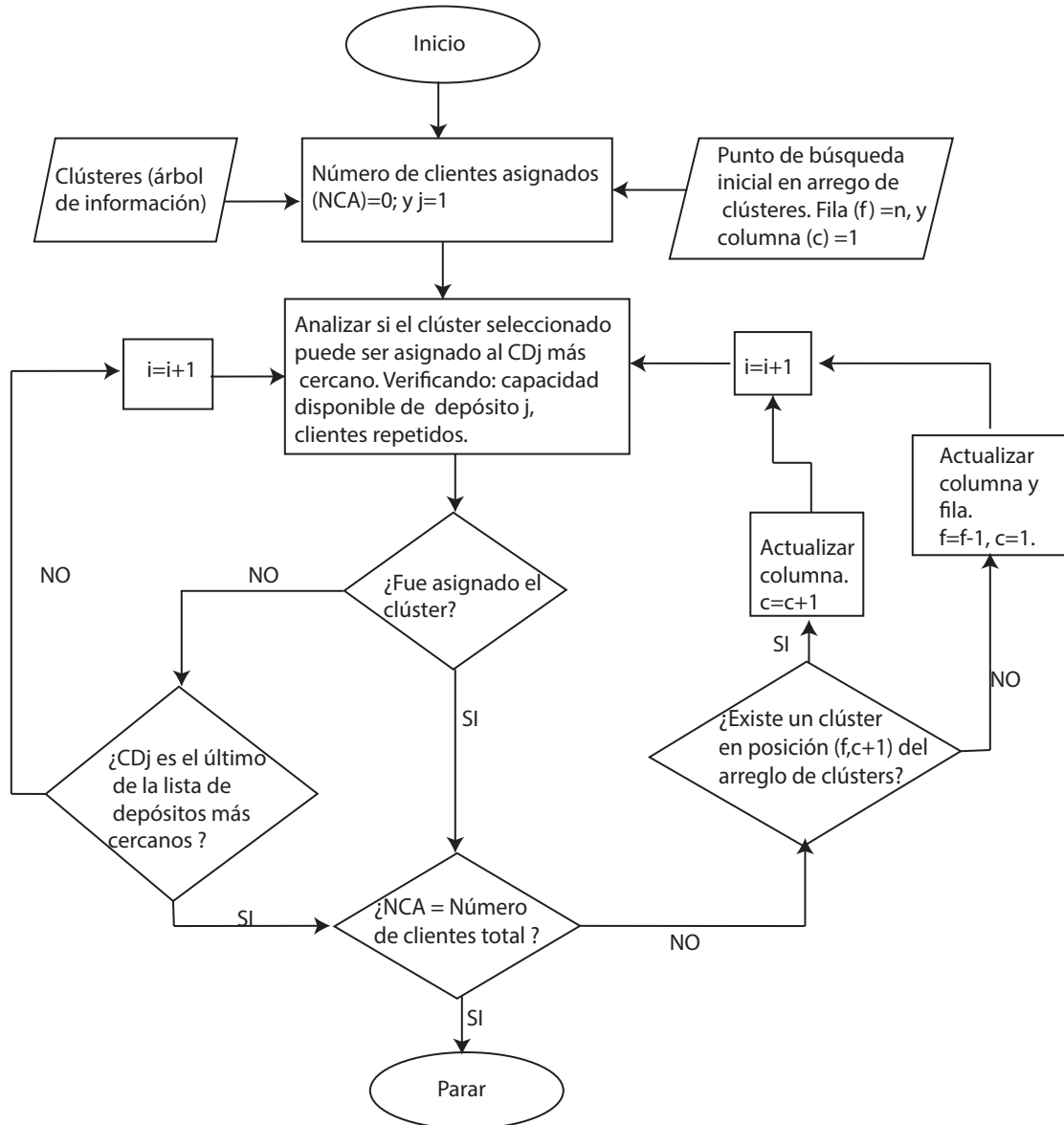


Figura 3.11: Algoritmo de asignación de clientes a los depósitos.

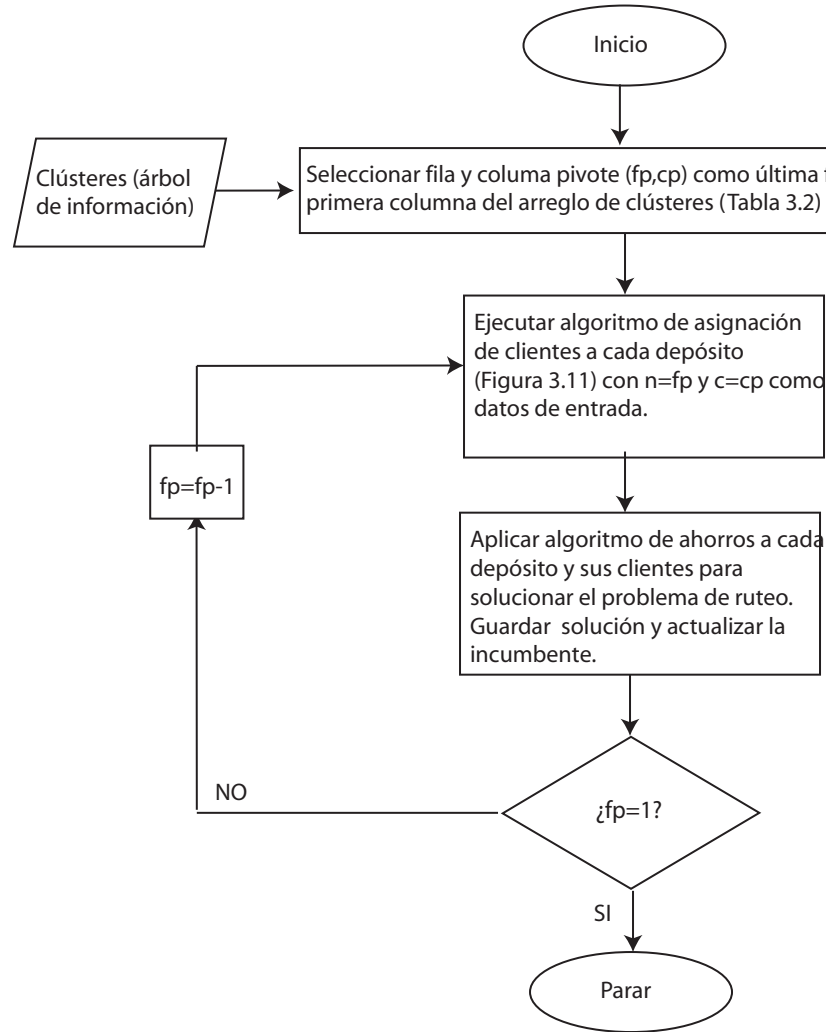


Figura 3.12: Algoritmo de generación de soluciones iniciales.

3.6. Algoritmo de Búsqueda Local Iterada (ILS)

El algoritmo de búsqueda local consiste básicamente en un procedimiento de intensificación alrededor de la solución actual y se realiza identificando soluciones vecinas y para esto se usa una estructura de vecindad. Esta estructura se logra efectuando cambios

a la solución actual, que después de ser evaluados se selecciona la de mejor calidad. En este algoritmo la estructura de vecindad se logra haciendo uso de operadores intra e inter-ruta. Con los primeros se efectúa búsqueda local próxima y con los segundos se logra llegar a un vecindario de segundo nivel, de esa forma se cuenta con intensificaciones entre baja y alta intensidad. En la sección 3.3. se presentan los operadores inter e intra-ruta. En las figuras 3.3 a 3.8 se muestra su representación gráfica. En la primera etapa se identifican soluciones iniciales de buena calidad y en bajos tiempos de cómputo y para lograrlo se usa una heurística constructiva que emplea criterios de clusterización y operadores inter e intra-ruta. La segunda etapa del proceso es inicializada con la mejor solución encontrada en la etapa 1. Esta etapa usa el algoritmo ILS y tiene por objetivo encontrar una o varias soluciones de excelente calidad.

El algoritmo propuesto se ejecuta de la siguiente manera:

1. Seleccionar un operador al azar de la lista de estructuras de vecindad inter-ruta. Si hay alguna mejora ir al paso 3, de lo contrario, ir al paso 2.
2. En caso de que la aplicación de la estructura de vecindad no pueda mejorar la solución incumbente, prohibirla para la próxima selección. Si todas las estructuras de vecindad han sido prohibidas, la búsqueda local se detiene. En caso contrario ir al paso 1.
3. Seleccionar una estructura de vecindad de la lista de operadores intra-ruta. Si hay una mejora, ir al paso 5. De lo contrario, ir al paso 4.
4. Si la aplicación de la estructura de vecindad dentro de la ruta no puede mejorar la solución actual, la estructura queda prohibida para las selecciones posteriores. Ir al paso 1 si se han prohibido todas las estructuras de vecindad intra-ruta. De lo contrario ir al paso 3.
5. Ir al paso 3.

La lista de estructuras de vecindad inter-ruta usada es : Shift(1,0), Shift(2,0), Swap(1,1), Swap(2,2), 2-OPT. [23]

La lista intra-ruta está formada por: Shift(1,0), Swap(1,1), 2-OPT. [23]

3.6.1. Procedimiento de perturbación en el algoritmo ILS

El procedimiento de perturbación se lleva a cabo con clientes seleccionados aleatoriamente, los cuales deben pertenecer a diferentes rutas. Los operadores usados para tal fin son: swap(1,1), swap(2,1), swap(2,2), shift(1,0), shift(2,0). El operador seleccionado se aplica dos veces y de forma consecutiva. Para aplicar el operador se debe cumplir que la distancia entre los clientes seleccionados, no sobrepasen los límites de una distancia que es calculada a cada paso. Este valor se calcula como el promedio de la distancia del depósito asociado al primer cliente, respecto a los dos depósitos más cercanos. Esa distancia es multiplicada por un factor β seleccionado aleatoriamente y de acuerdo a pruebas preliminares debe estar entre $[0.1 - 1.0]$.

Capítulo 4

Análisis de resultados

Para validar la metodología propuesta se utilizan las instancias presentadas por Cordeau et al. [33] planteadas para el MDVRP. Los valores de referencia para las instancias tipo P son presentadas en la Tabla 4.1, en la cual se observan las mejores soluciones reportadas (BKS) y las características generales de cada instancia como son: número de clientes, número de depósitos y número de vehículos.

Tabla 4.1: Instancias tipo (P) propuestas en literatura con su respectivo BKS.

Instancia	BKS	# Clientes	# Depósitos	# Vehículos
P01	576,86	50	4	4
P02	473,53	50	4	2
P03	640,65	75	5	3
P04	999,21	100	2	8
P05	750,03	100	2	5
P06	876,5	100	3	6
P07	881,97	100	4	4
P12	1318,95	80	2	5
P15	2505,42	160	4	5
P18	3702,85	240	6	5
P21	5474,74	360	9	5

Adicionalmente, la metodología desarrollada se prueba utilizando las instancias tipo Pr, a las cuales se aplica restricción de distancia (D_x) que consiste en que ninguna de las rutas

sobrepase dicho valor. La Tabla 4.2 presenta los detalles de las instancias con restricción de distancia que fueron generadas por Cordeau et al. en [33].

Tabla 4.2: Instancias tipo (Pr) propuestas en literatura con su respectivo BKS.

Instancia	BKS	# Clientes	# Depósitos	# Vehículos	D_x
pr01	861,32	48	4	2	500
pr02	1296,25	96	4	4	480
pr03	1803,80	144	4	6	460
pr04	2042,45	192	4	8	440
pr05	2324,12	240	4	10	420
pr06	2663,56	288	4	12	400
pr07	1075,12	72	6	3	500
pr08	1658,23	144	6	6	475
pr09	2131,70	216	6	9	450
pr10	2805,53	288	6	12	425

Los resultados serán analizados así: Inicialmente, se estudiara y analizará la eficiencia de las diferentes técnicas de clusterización utilizados como inicializadores, en las instancias tipo P; de donde se selecciona una de las técnicas de clusterización tomando como base para dicha selección el tiempo de computo y GAP de cada una de ellas. Seguidamente, será evaluado el desempeño del algoritmo ILS medido en la calidad de solución y tiempo de computo. Se tomaran como referencia los resultados presentados por Vidal et.al en [31], para los dos tipos de instancias Tipo P y Pr.

4.1. Resultados etapa de inicialización

Inicialmente se realiza una comparación de los diferentes tipos de clústers, mediante las heurísticas de clusterización que fueron descritas anteriormente. El algoritmo se implementa en C++ y ejecuta en un computador Intel i5-3470 3,2 GHZ, 8 GB. Para encontrar los parámetros de mejor desempeño se realizan pruebas intensivas iniciales. El número de iteraciones globales definido como criterio de parada del algoritmo es de 700.000. El factor β de mejor desempeño se obtiene mediante la generación de un valor aleatorio entre 0,1 y 1. En el proceso de perturbación, debido a que es necesario identificar el cliente 2 que cumpla las condiciones de permanencia a una ruta diferente de la del cliente 1, se realizan hasta 50

intentos.

Tabla 4.3: Resultados del método de inicialización usado.

Instancia	Paso de inicialización (%GAP)							
	WCM	CMU	AL	UAL	SL	CL	W	Min
P01	6,92	5,39	6,92	6,92	5,67	6,92	6,92	5,39
P02	6,66	6,25	6,66	5,72	6,66	6,03	6,66	5,72
P03	7,77	6,62	8,93	8,93	9,27	8,7	8,7	6,62
P04	7,99	7,99	7,99	7,99	6,62	8,85	8,85	6,62
P05	9,22	8,12	10,37	9,22	11,02	8,73	10,37	8,12
P06	7,96	8,02	6,56	8,02	6,55	5,76	7,64	5,76
P07	6,2	6,2	6,2	6,2	6,2	6,2	6,2	6,2
P12	3,54	3,54	2,12	1,61	3,54	1,07	3,1	1,07
P15	6,36	8,13	8,15	7,1	9,02	7,71	6,21	6,21
P18	9,49	7,33	10,08	7,45	10,65	10,47	8,94	7,33
P21	10,95	9,27	10,44	9,63	12,25	11,22	11,85	9,27
Media	7,55	6.99	7,67	7,16	7,95	7,42	7,7	6,99

Los resultados de la Tabla 4.3 presentan las mejores soluciones encontradas al realizar 10 ejecuciones del algoritmo para cada una de las heurísticas de clusterización. Como se observa, las soluciones iniciales encontradas son similares respecto al valor de la función objetivo reportada. También se puede observar como cada técnica heurística de clusterización y para cada una de las instancias se identifica respuestas en diferentes espacios de solución, lo que será corroborado al momento de aplicar el algoritmo de optimización ILS. La calidad de este espacio es determinante tanto en la calidad de la solución final como en el tiempo de cómputo requerido.

Se puede observar que se realiza énfasis en el GAP más bajo encontrado por cada técnica. La CMU determina el mejor GAP para 5 instancias, obteniendo un valor de GAP medio de 6.99 y posicionando como el mínimo entre todo los métodos de clusterización, por lo tanto se puede concluir que haciendo referencia al GAP la técnica CMU corresponde a la de mejor desempeño.

En el caso de la instancia PO7, todas las técnicas heurísticas de clusterización identifican la misma solución inicial, por lo tanto, se puede concluir que todas las técnicas encuentran siempre la misma solución.

En la Tabla 4.4 se presentan los resultados de las heurísticas de clusterización en relación a tiempos de cómputo, notando que W es la que tarda mayor tiempo y sería inviable desde el punto de vista computacional. Por otro lado, CMU alcanza con más frecuencia los mejores tiempos aunque no es la de mejor tiempo promedio.

Tabla 4.4: Tiempos computacionales en los métodos de inicialización usados para cada instancia.

Instancia	Tiempo computacional en la etapa de inicialización (segundos)						
	WCM	CMU	AL	UAL	SL	CL	W
P01	0,07	0,071	0,075	0,076	0,127	0,108	0,711
P02	0,097	0,095	0,102	0,112	0,169	0,148	0,718
P03	0,268	0,299	0,352	0,281	0,603	0,471	7,988
P04	0,624	0,624	0,734	0,655	1,279	0,951	40,17
P05	0,627	0,671	0,849	0,697	1,659	1,154	40,139
P06	0,593	0,608	0,655	0,609	1,233	0,936	40,092
P07	0,764	0,752	1,036	0,832	1,886	1,142	46,254
P12	0,343	0,312	0,359	0,375	0,905	0,546	8,799
P15	2,777	2,668	2,762	2,636	12,939	4,845	526,126
P18	13,039	12,496	14,619	9,033	70,832	17,737	6685,146
P21	41,075	58,422	40,82	47,421	327,467	57,926	69878,787
Media	5,48	7	5,67	5,7	38,1	7,81	7024,99

Con el fin de estudiar el desempeño de las heurísticas de clusterización, al ser usadas como soluciones iniciales, fue implementado un algoritmo ILS básico. Con base en este se realizaron 10 corridas, para cada uno de los casos y para cada una de estas, es reportado el promedio y mejor solución, la que se indica con (*). En la Tabla 4.5 se presentan los resultados. El tiempo de cálculo más bajo se obtiene cuando se utiliza CMU como solución inicial. En la mayoría de los casos en los que se utiliza esta técnica de clusterización como estrategia de inicio se puede ver que las soluciones son de buena calidad dado el valor de GAP bajo y el tiempo computacional requerido. Comparando con los resultados encontrados en la etapa de inicialización (Tabla 4.3), se observa que se mantiene la relación de que una buena respuesta en la etapa de inicialización garantiza buenas respuestas en el desarrollo del algoritmo ILS.

En la Tabla 4.6 se presenta el tiempo total requerido para la ejecución del algoritmo, que incluye la técnica de inicialización mediante las métricas de clusterización y el algoritmo de ILS. Es de observar que en la fase de optimización se obtienen soluciones de diferente

Tabla 4.5: Resultados del algoritmo ILS comparado con las BKS.

Instancia	Resultados con el algoritmo ILS (%GAP)													
	WCM*	WCM	CMU*	CMU	AL*	AL	UAL*	UAL	SL*	SL	CL*	CL	W*	W
P01	0	2,2	0	1,16	0	1,16	0	0,98	4,94	5.27	0	1.07	0	2.16
P02	0,59	3,43	0,85	1,82	0,59	3,14	0,85	2,25	1,25	3.44	0,85	1.91	0,59	3.48
P03	0	1,86	0	1,27	0	1,99	0	1,7	0	0.98	0,62	1.26	0	0.94
P04	0,05	1,67	0,026	2,11	0	1,48	0,26	1.52	0,27	4.68	0,05	1.73	0,45	2.22
P05	0,15	0,19	0,15	0,27	0	0,26	0,15	0,22	0,15	0.19	0,07	0.24	0	0.26
P06	0,87	2,18	0	5,42	0,09	1,24	1,93	5,17	0,45	1.36	0,7	2.10	0,33	1.86
P07	2,68	4,79	2,68	4,62	2,68	4,91	2,68	5,03	2,68	5.02	2,68	5.02	2,68	4.60
P12	0	0	0	0	0	0,6	0	0,33	0	0.00	0	0.00	0	0.60
P15	0	2,59	0	1,87	0	2,57	0	2,62	0	2.01	0	1.92	0	1.68
P18	0	3,17	0,16	4,05	0,7	4,18	1,55	4,3	0	2.60	0	2.91	0	0.00
P21	0,002	3,35	0,46	4,7	0,39	4,86	2,3	6,57	0,63	3.35	1,75	4.82	0,98	0.00
Media	0,39	2,31	0,39	2,48	0,4	2,4	0,88	2,79	0,94	2.63	0,61	2.09	0,46	1.98

calidad cuando son usadas diferentes heurísticas de inicialización, con las cuales se identifican soluciones en diferentes puntos del espacio de soluciones. Este aspecto es conveniente si lo que se desea es realizar es una búsqueda diversa, en el espacio de soluciones, ya que permite que el algoritmo arranque su operación en diferentes puntos y permita una mejor exploración de dicho espacio.

Comparando los BKS con las soluciones iniciales encontradas con técnicas de clusterización, se observa que estas técnicas localizan los clientes al depósito más cercano, en un porcentaje superior al 80% en todos los casos, lo que demuestra la fortaleza de las mismas. Además comparado con la Tabla 4.4, se observa que a pesar de que en la etapa de inicialización el método WCM tiene un menor tiempo promedio resultante, cuando se combina con el ILS la técnica CMU, se comporta mejor respecto a tiempos computacionales. Por lo tanto, aunque un inicializador demore más en dar una buena solución, es un tiempo muy bajo, cuando es comparado con el requerido por el algoritmo en la etapa de optimización, al ser aplicado el algoritmos ILS.

Tabla 4.6: Metodología completa.

instancia	Tiempo computacional de inicialización + ILS (segundos)						
	WCM	CMU	AL	UAL	SL	CL	W
P01	0,279	0,223	0,28	0,296	0,437	0,318	0,96
P02	103,172	39,421	99,752	102,141	2,545	2,486	100,792
P03	1,048	18,691	13,6	13,369	1,651	2,6	9,485
P04	122,363	74,007	65,625	136,781	60,544	53,43	236,341
P05	51,448	7,582	193,63	43,597	140,447	382,044	164,174
P06	147,514	187,949	161,881	25,657	17,02	113,35	46,129
P07	30,61	30,185	30,64	30,646	31,252	30,379	76,191
P12	0,343	0,312	0,858	4,602	0,905	0,578	15,928
P15	155,625	22,76	73,851	95,144	77,384	327,792	570,805
P18	3534,845	677,649	2025,411	72,899	3463,423	620,011	7995,713
P21	8567,631	846,753	7504,1	5576,696	6237,983	995,848	72951,8694
Media	1155,9	173,23	924,51	554,71	912,14	229,89	7469,85

Se puede concluir que las técnicas de clusterización usadas como inicializadores facilitan el desarrollo de la técnica ILS para encontrar soluciones de alta calidad en tiempos computacionales razonables. La afirmación anterior se debe a que el algoritmo de optimización presenta un buen rendimiento, ya que pasa de 6.21 % en la etapa inicial hasta 0.30 % de GAP en la etapa de optimización. Para los 11 casos analizados se obtuvo el BKS en 6 de ellos. Para P21, la instancia más grande, se obtuvo una configuración óptima totalmente diferente a la reportada en la literatura.

4.2. Resultados del algoritmo ILS usando instancias tipo P para el MDVRP

Utilizando como heurística de inicialización el método CMU y como algoritmo de optimización la metaheurística ILS, son resueltas las instancias propuestas por Vidal et al. en [31]. Los resultados son presentados en la Tabla 4.7 y se reporta el mejor valor de la función objetivo, después de corridas 10 veces la misma instancia. Así como el promedio de la función objetivo, también es reportado el tiempo requerido y cada instancia utiliza un total de 70,000 iteraciones. Además en la misma tabla se muestra el BKS para cada una de las instancias,

así como el GAP medido como la diferencia porcentual entre el BKS y la mejor solución obtenida con la metodología propuesta; el tiempo está dado en minutos.

Tabla 4.7: Resultados usando el algoritmo ILS y el inicializador CMU, comparados con los presentados en [31].

Instancia	Metodología Propuesta			Vidal		BKS	Gap %
	Promedio	Tiempo	Fobj	Tiempo	Fobj		
P01	576,87	0,22	576,87	0,49	576,87	576,87	0
P02	473,53	0,23	473,53	0,81	473,53	473,53	0
P03	641,18	0,62	641,18	0,99	640,65	640,65	0,08
P04	1004,61	1,76	1001,03	0,81	999,21	999,21	0,18
P05	750,93	2,68	750,02	2,09	750,03	750,03	0
P06	882,76	1,88	878,1	1,6	876,5	876,5	0,18
P07	887,45	3,56	884,66	1,63	881,97	881,97	0,31
P12	1318,95	0,01	1318,95	1,09	1318,95	1318,95	0
P15	2505,42	2,46	2505,42	3,25	2505,42	2505,42	0
P16	2597,29	6,17	2597,29	3,15	2572,33	2572,23	0,97
P17	2722,77	0,01	2722,77	3,03	2709,09	2709,09	0,5
P18	3710,69	11,11	3707,77	10,32	3702,85	3702,85	0,13

Como se puede observar, se alcanza la mejor solución reportada para 5 instancias en menor tiempo computacional, además para las otras instancias la metodología muestra dar soluciones de buena calidad y la desviación de esta respecto al promedio de la encontrada en las 10 ejecuciones es baja. También se observa que los tiempos computacionales son muy competitivos respecto a los reportados en la literatura.

4.3. Resultados del algoritmo ILS para instancias del MDVRP con restricción de distancia

Como en el caso anterior, se usa como heurística de inicialización el método CMU. Para fines de comparar resultados se usan los encontrados por Vidal et al. en [31]. Para cada instancia se utiliza un total de 70,000 iteraciones y cada una de las instancias es corrida 10 veces. Para la metodología propuesta se presenta el promedio de los 10 casos de cada una de las instancias estudiadas. Además se reporta la de mejor valor (Fobj) y el tiempo requerido

para su ejecución, esta dado en minutos. El (Gap %) es calculado mediante la diferencia porcentual entre la Fobj y el BKS. Las instancias Pr, a diferencia de las tipo P, tienen como restricción un parámetro asociado a la distancia máxima permitida en la ruta.

Tabla 4.8: Resultados usando el algortimo ILS y el inicializador CMU y resultados de la referencia [31] , con restricción de distancia.

Instancia	Metodología Propuesta			Vidal		BKS	Gap %
	Promedio	Tiempo	Fobj	Tiempo	Fobj		
Pr01	861,32	0,21	861,32	1,02	861,32	861,32	0
Pr02	1296,25	1,55	1296,25	2,82	1296,25	1296,25	0
Pr03	1803,80	5,28	1803,80	5,66	1803,80	1803,80	0
Pr04	2076,72	8,22	2069,76	8,09	2045,45	2042,45	1,34
Pr05	2357,16	18,00	2350,69	11,41	2326,35	2324,45	1,13
Pr06	2715,26	48,56	2704,50	18,06	2668,76	2663,56	1,54
Pr07	1077,33	0,27	1077,33	1,55	1075,12	1075,12	0,21
Pr08	1658,23	5,56	1658,23	4,30	1658,23	1658,23	0
Pr09	2167,93	8,67	2154,82	8,85	2131,70	2131,70	1,08
Pr10	2911,79	34,23	2911,79	17,96	2810,25	2805,53	3,79

Como se observa en la Tabla 4.8, a pesar de no superar los BKS reportados en la literatura, se logra alcanzar el GAP en cuatro de las instancias. Para las demás instancias se logran soluciones muy cercanas a las reportadas en la literatura. Los tiempos de cómputo se muestran muy competitivos respecto a los registrados por Vidal et al. [31].

Capítulo 5

Conclusiones y trabajo futuro

- Se propone una nueva metodología de inicialización utilizando diferentes técnicas de agrupamiento, desarrolladas para las instancias reportadas en la literatura, alcanzando soluciones de buena calidad con un GAP (%) bajo respecto a la mejor solución reportada, el tiempo de computo más bajo se obtiene utilizando el método CMU como estrategia de inicialización.
- Después de aplicar las métricas de clusterización, se propone un mecanismo eficiente de exploración de la estructura tipo árbol (ILS), que permite resolver el MDVRP para las instancias clásicas y con restricción de distancia, a pesar de que ninguna instancia es superada en su función objetivo, se alcanza la mejor solución encontrada en muchas de ellas en menores tiempos de computo.
- Encontrar diferentes soluciones de inicio de buenas calidad mediante las técnicas de clusterización, permite que el algoritmo ILS inicie de puntos diversos y lo favorezca hacia una mejor exploración en el espacio de solución del problema.
- Se destaca de acuerdo a la calidad de solución y tiempo de computo, las métricas CMU, W, AL y WCM, en el orden en que son numeradas. La técnica CMU, es utilizada para comparar los resultados obtenidos con los más actuales reportados.
- La metodología planteada con base en el algoritmo ILS y la heurística de inicialización

CMU, logra GAP 0 % en 5 de las instancias tipo P y en 4 de las instancias tipo Pr. En tiempos de cómputo que en unos casos son mas bajos y en otros próximos a los reportados. En las demás instancias las respuestas obtenidas son muy próximas a los BKS reportados en la literatura, por tal razón de la metodología propuesta se concluye que obtiene soluciones de muy buena calidad al ser comparadas con las referenciadas en la literatura y en tiempos de cómputo competitivos.

- Se propone como trabajo futuro, implementar una metaheurística poblacional empleando diferentes estrategias de clusterización para generar las soluciones de inicio. Además, esta propuesta puede ser extendida a otros problemas de ruteo comunes en la literatura, utilizando las diferentes métricas descritas.

Bibliografía

- [1] R. Dantzig, The Truck Dispatching Problem, Management Sci, vol. 6, n 1, pp 80.
- [2] J. Montoya, J. López, S. Nieto, N. Herazo, A literature review on the vehicle routing problem with multiple depots, Computer & Industrial Engineering, 79 (2015) 115-129.
- [3] Kulkarni R.V & Bhavne, P.R, Integer programming formulation of vehicle routing problem. European journal of Operational Research, (1985) 20(1), 58-57.
- [4] Laporte, G., integer programming formulation for the multi-depot vehicle routing problem. Comments on a paper by Kulkarni and Bhavne. European Journal of Operational Research, (1989), 38(2), 228-237.
- [5] Carpaneto, G., Dell'Amico, M., Fischetti, M., & Toth, P., A branch and bound algorithm for the multiple depot vehicle routing problem. Networks, (1989), 19(5), 531-548.
- [6] Renaud, J., Boctor, F.F., & Laporte, G. A tabu search heuristic for the multidepot vehicle routing problem. Computer & Operations Research, (1996), 23(3), 229-335.
- [7] Christofides, N., & Eilon S. An algorithm for the vehicle- dispatching problem. Journal of the Operational Research Society, (1969), 20, 309-318.
- [8] Gillet, B., & Johnson, J. Multi-terminal vehicle dispatch algorithm. Omega, (1976), 4(6), 711-718.
- [9] Tüzün, D., Burke, L. A two-phase tabu search approach to the location routing problem. European Journal of Operational Research, (1999), 117(2), 483-495. .

- [10] Cordeau, J. F., Laporte, G., & Mercier, A. A unified Tabu search Heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, (2001), 52 (8), 928-936.
- [11] Filipec, M., Skrllec, D., & Krajcar, S. Darwin meets computers: new approach to multiple depot capacitated vehicle routing problem. In *Proceeding of the 1997 IEEE international conference on systems, man, and cybernetics*, Vol.1, (1997) 421-426.
- [12] Vianna, D. S., Ochi, L. S., & Drummond, L.M. A parallel hybrid evolutionary metaheuristic for the period vehicle routing problem. *Parallel and distributed processing. Lecture notes in computer science*. Vol 1586, (1999), Berlin.
- [13] Thangiah, S.R., & Salhi, S. Genetic clustering: An adaptive heuristic for the multi-depot vehicle routing problem. *Applied Artificial Intelligence*. (2001), 15(4), 361-383.
- [14] Lee, M. a. Epelman, C. C. A shortest path approach to the multiple-vehicle routing problem with split pick-ups. *Transp.Res.Part B Methodol.*, Vol.40, no.4, pp. 265-284, 2006.
- [15] B. Crevier, J.F. Cordeau, & G. Laporte. The multi-depot vehicle routing with inter-depot routes. (2007), *European Journal of Operational Research*, vol 176, pp 756-773.
- [16] G. Jeon, H.R. Lepp & J.Y. Shim, a vehicle routing problem solved by using a hybrid genetic algorithm, (2007) *Comput.Ind.Eng.*, Vol. 53, no.4, pp. 680-692.
- [17] W. Ho, G. T. S. Hoo, P. Ji, a hybrid genetic algorithm for the multi-depot vehicle routing problem, (2008), *Eng. Appl. Artif. Intell.*, vol. 21, no. 4, pp. 548-557.
- [18] A. Subramanian, "Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems," *Universidade Federal de Fluminense*, 2012.
- [19] H.-k.Chen, C.-F. Hsueh, and M.-S.Chang, "the real-time time-dependent vehicle routing problem," *Transp. Res. Part E logist. Transp. Rev.*, vol. 42, no. 5, pp. 383-408, Sep. 2006.
- [20] R. Dantzig, The Truck Dispatching Problem, *Management Sci*, vol. 6, n 1, pp 80.

- [21] A. C. Rencher, A Review Of "*Methods of Multivariate Analysis*", vol- 37, no. 11. 2005.
- [22] I. D. Giosa, I. L. Tansini, and I. O. Viera, "New assignment algorithms for the multi-depot vehicle routing problem," *J. Oper. Res. Soc.*, vol 53, no.9, pp. 977-984, Sep 2002.
- [23] S. Barreto, C. Ferreira, J. Paixão, and B. S. Santos, "Using clustering analysis in a capacitated location-routing problem," *Eur. J. Oper. Res.*, vol. 179, no. 3, pp. 968–977, Jun. 2007.
- [24] H.R. Lourenco, O. Martin, and T. Stützle. Handbook of Metaheuristics, chapter Iterated Local Search, Kluwer Academic Publishers, 321-353, 2002.
- [25] T. Stützle. Local search algorithms for combinatorial problems analysis, algorithms and new applications. Technical report, DISKI Dissertationen zur Künstlichen Intelligenz. Sankt Augustin, Germany, 1999.
- [26] T. Stützle. Local Search Algorithms for Combinatorial Problems Analysis, Improvements and New Applications. PhD Thesis, Darmstadt, University of Technology, Department of Computer Science, 1998.
- [27] Subramanian, A. Heuristic, exact and hybrid approaches for VRP. PhD Thesis. Universidad Federal Fluminense. Niteroi (2012).
- [28] Anderberg, M. R. (2014). Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks (Vol. 19). Academic press.
- [29] Edelbrock, C., & McLaughlin, B. (1980). Hierarchical cluster analysis using intraclass correlations: A mixture model study. *Multivariate Behavioral Research*, 15(3), 299-318.
- [30] Dubes, R., & Jain, A. K. (1980). Clustering methodologies in exploratory data analysis. *Advances in computers*, 19, 113-228.
- [31] Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2014). Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research*, 237(1), 15-28.

- [32] P. Surekha & S. Sumathi. Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms. *World Applied Programming*, (August):118–131, 2011.
- [33] J.-F. Cordeau, M. Gendreau, & G Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:1052119, 1997
- [34] R. Baldacci and A. Mingozzi. A uniVed exact method for solving diUerent classes of vehicle routing problems. *Mathematical Programming*, 120:347–380, 2009. ISSN 00255610.
- [35] T. Vidal, T. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research*, 60(3):611–624, 2012.
- [36] C. Contardo and R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014.
- [37] Escobar, J. W., Linfati, R., Toth, P., & Baldoquin, M. G. (2014). A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. *Journal of heuristics*, 20(5), 483-509.
- [38] Xu, D., & Xiao, R. (2015). An improved genetic clustering algorithm for the multi-depot vehicle routing problem. *International Journal of Wireless and Mobile Computing*, 9(1), 1-7.
- [39] de Oliveira, F. B., Enayatifar, R., Sadaei, H. J., Guimarães, F. G., & Potvin, J. Y. (2016). A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem. *Expert Systems with Applications*, 43, 117-130.
- [40] Tlili, T., Krichen, S., Drira, G., & Faiz, S. (2016). On Solving the Multi-depot Vehicle Routing Problem. In *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics* (pp. 103-108). Springer, New Delhi.

- [41] Sadykov, R., Pessoa, A. A., & Uchoa, E. (2017, July). A branch-cut-and-price algorithm for the distance constrained multi-depot vehicle routing problem. In 21st Conference of the International Federation of Operational Research Societies.
- [42] Ospina-Toro, D., Toro-Ocampo, E. M., & Gallego-Rendón, R. A. (2018). Solución del MDVRP usando el algoritmo de búsqueda local iterada. *Revista Colombiana de tecnologías avanzada (RCTA)*, 1(31).
- [43] Yuan, W., Wang, J., Li, J., Yan, B., & Wu, J. (2017, September). Two-Stage Heuristic Algorithm for a New Model of Hazardous Material Multi-depot Vehicle Routing Problem. In *UK Workshop on Computational Intelligence* (pp. 362-366). Springer, Cham.
- [44] Wassan, N., Wassan, N., Nagy, G., & Salhi, S. (2017). The multiple trip vehicle routing problem with backhauls: formulation and a twolevel variable neighbourhood search. *Computers & Operations Research*, 78, 454-467.
- [45] Soto, M., Sevaux, M., Rossi, A., & Reinholz, A. (2017). Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem. *Computers & Industrial Engineering*, 107, 211-222.
- [46] Zhang, S., Zhang, W., Gajpal, Y., & Appadoo, S. S. (2019). Ant Colony Algorithm for Routing Alternate Fuel Vehicles in Multi-depot Vehicle Routing Problem. In *Decision Science in Action* (pp. 251-260). Springer, Singapore.
- [47] Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*.
- [48] Everitt, B. (1974). *Cluster analysis* 122.
- [49] Mehrjerdi, Y. Z., & Nadizadeh, A. (2013). Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands. *European Journal of Operational Research*, 229(1), 75-84.

- [50] Toro-Ocampo, E. M., Domínguez-Castaño, A. H., & Escobar-Zuluaga, A. H. (2016). Desempeño de las técnicas de agrupamiento para resolver el problema de ruteo con múltiples depósitos. *Tecno Lógicas*, 19(36), 49-62.